Cryptographic Foundations
of
Blockchains

Hong-Sheng Zhou
Virginia Commonwealth University
http://www.people.vcu.edu/~hszhou/
hongsheng.zhou@gmail.com

Blockchain Summer School, Beijing, Aug 14-18, 2017

Plan

- Bitcoin basics
- Cryptographic foundations
- Nakamoto’s protocol
- Alternative mechanisms
- Crypto on the blockchain

many of the slides are from
Professors Aggelos Kiayias, Roger Wattenhofer, and Vassilis Zikas

Bitcoin Blockchain

How Bitcoin works under the hood

The Bank of Bitcoin

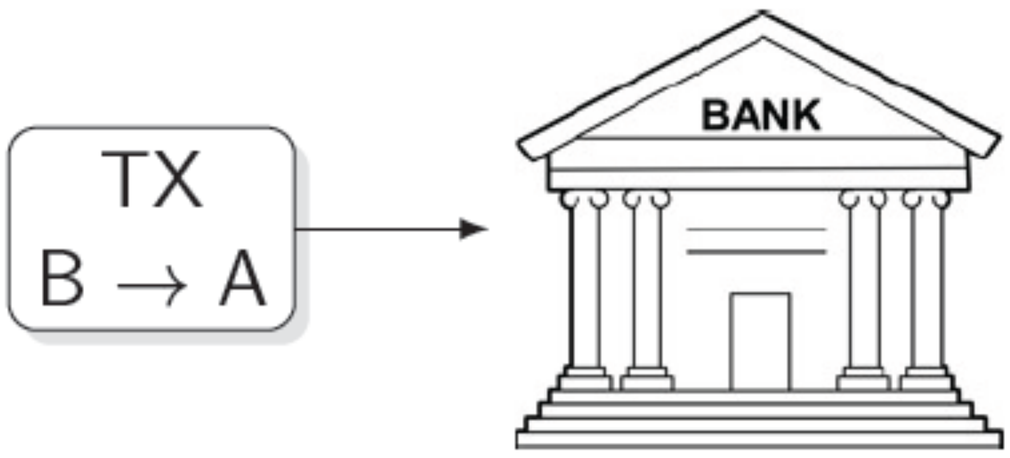
The Bank of Bitcoin

User	Balance
A	2
B	5
C	8

ledger

a ledger for all user activities

The Bank of Bitcoin

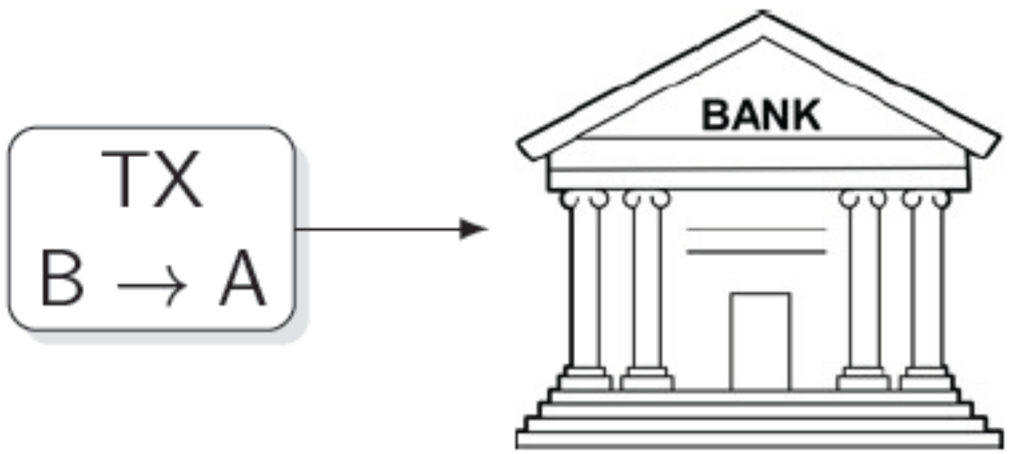


User	Balance
A	2
B	5
C	8

ledger

a ledger for all user activities

The Bank of Bitcoin



User	Balance
A	~~2~~ 4
B	~~5~~ 3
C	8

ledger

a ledger for all user activities

Opening an Account in Bitcoin

User

cryptographic tool:

digital signature scheme

private key for signature generation

public key for signature verification

public key = user account address

Transferring Bitcoins

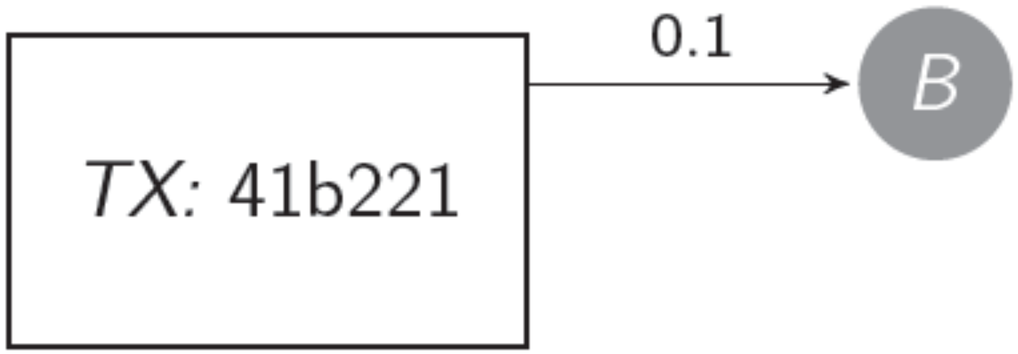
User

TX: 41b221

user A transfers 0.01 coins to user B

Transferring Bitcoins

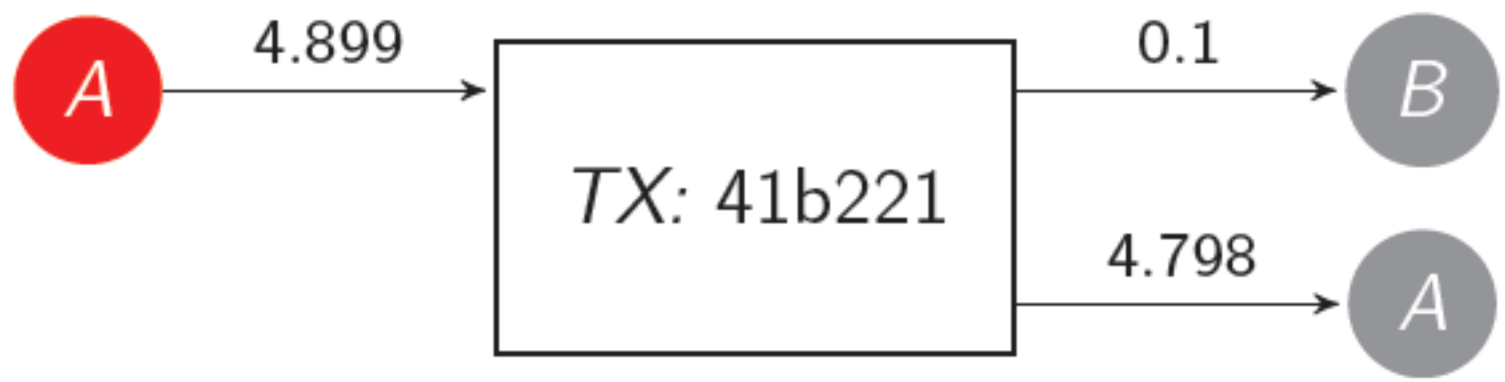
User



user A transfers 0.01 coins to user B

Transferring Bitcoins

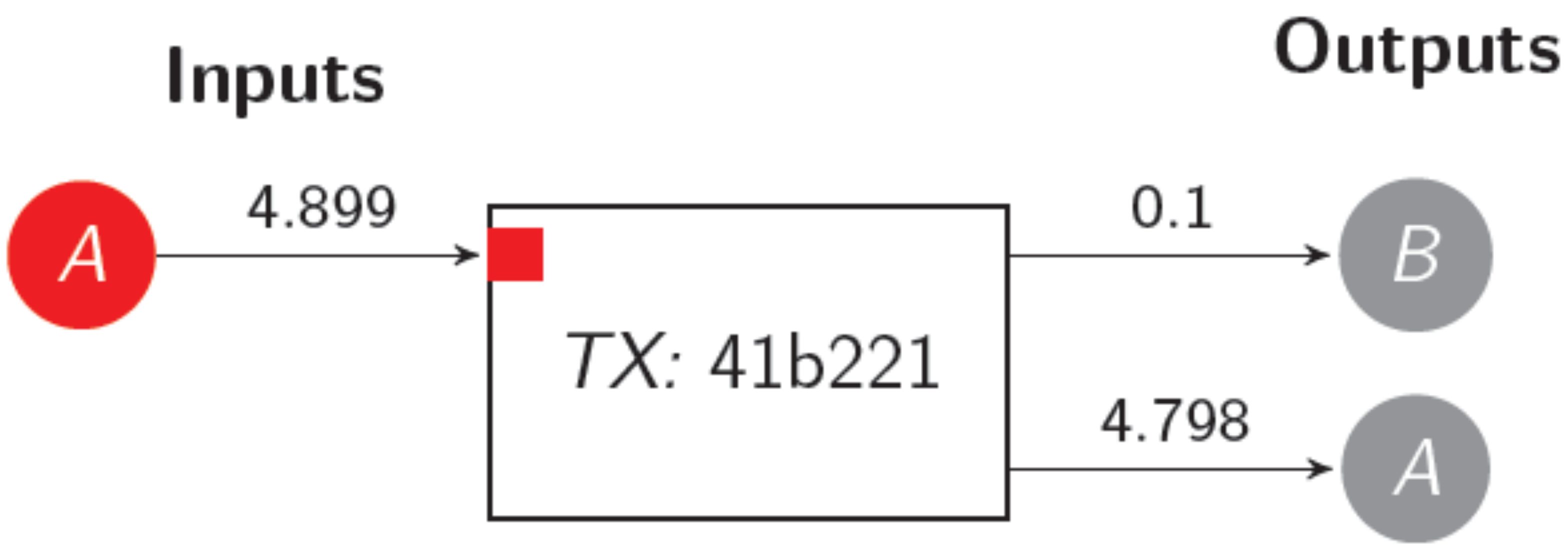
User



user A transfers 0.01 coins to user B

Transferring Bitcoins

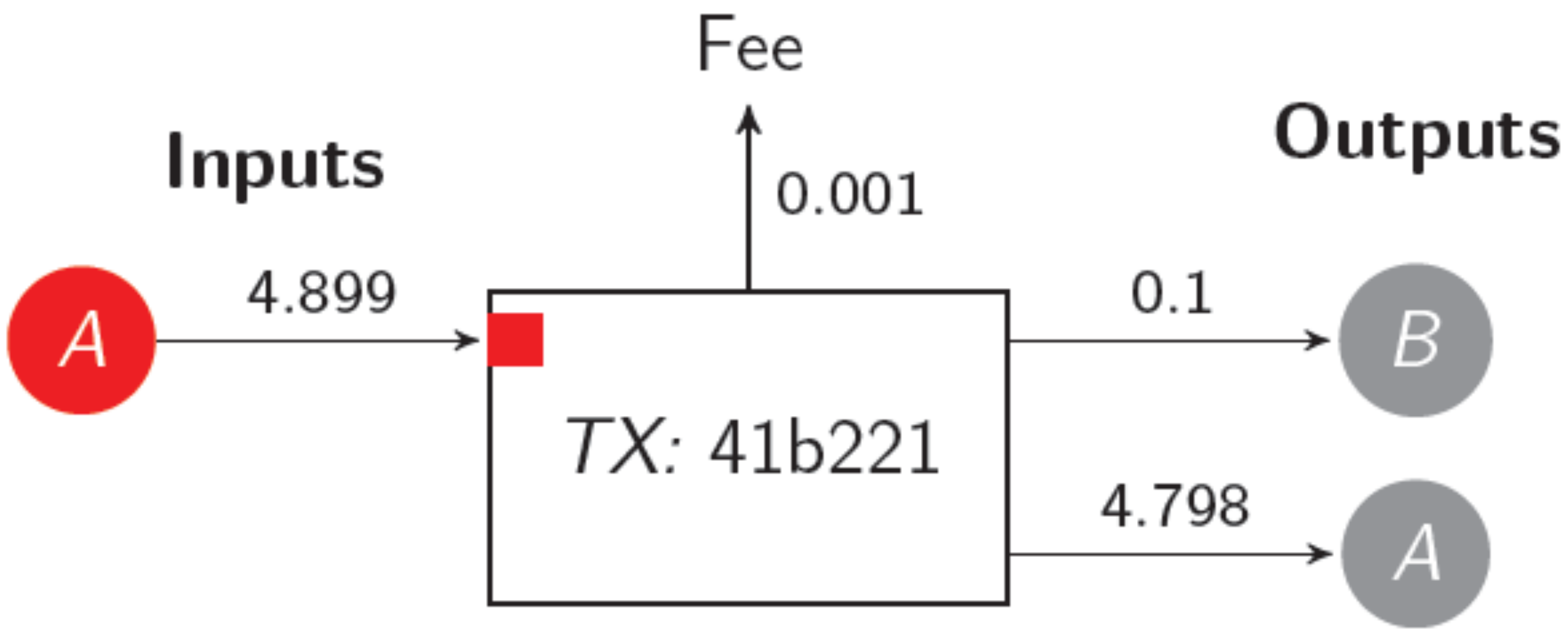
User



user A transfers 0.01 coins to user B

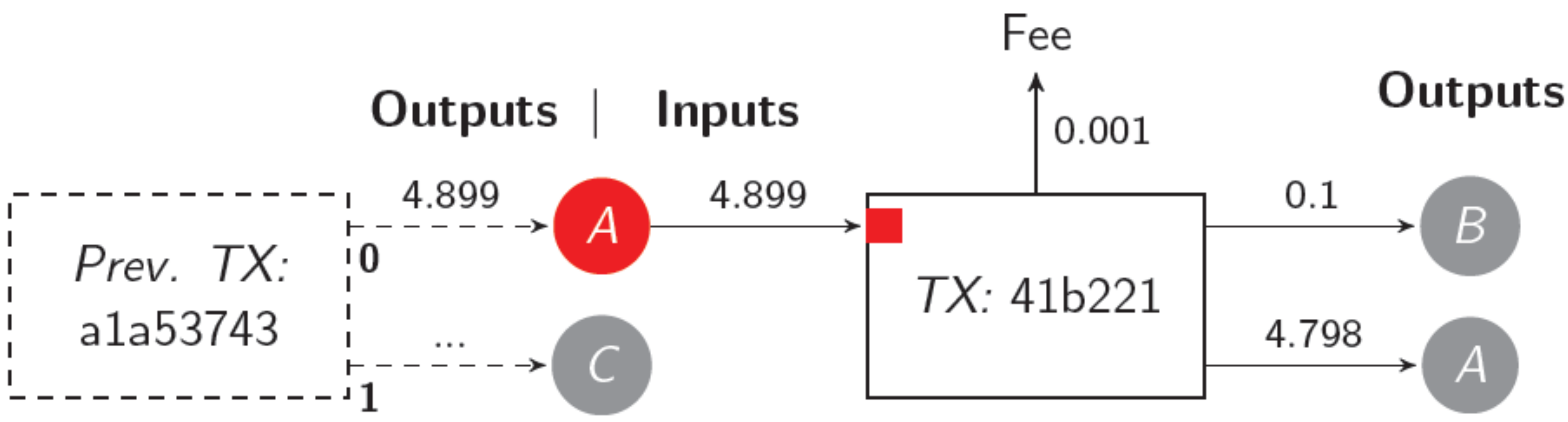
Transferring Bitcoins

User



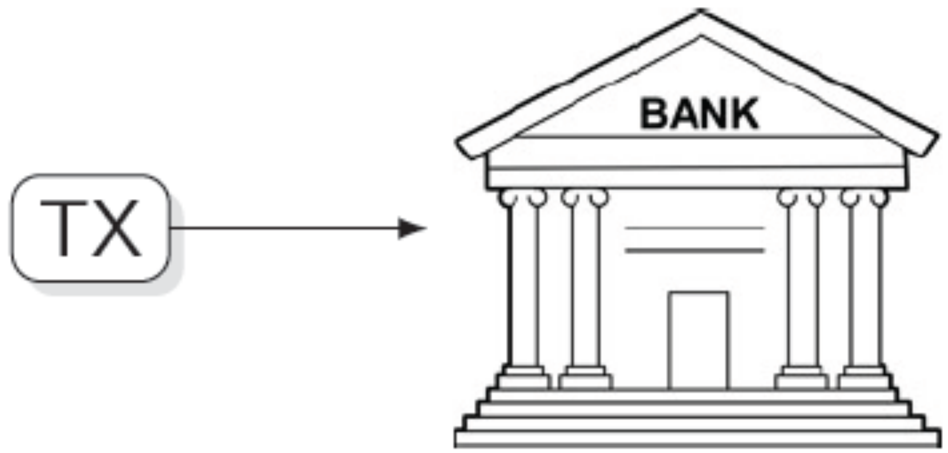
user A transfers 0.01 coins to user B

Transferring Bitcoins



user A transfers 0.01 coins to user B

Distributing the Bank

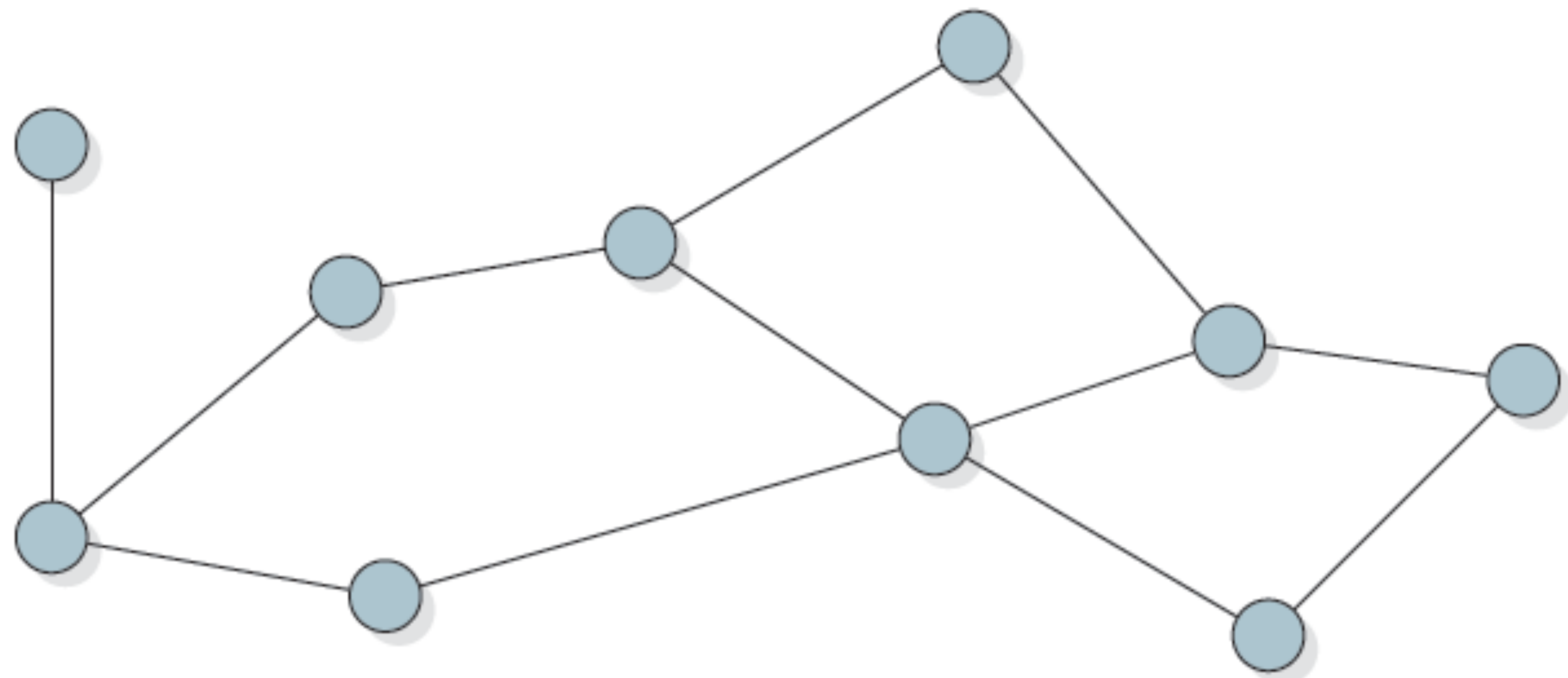


User	Balance
A	2
B	5
C	8

ledger

distribute the ledger

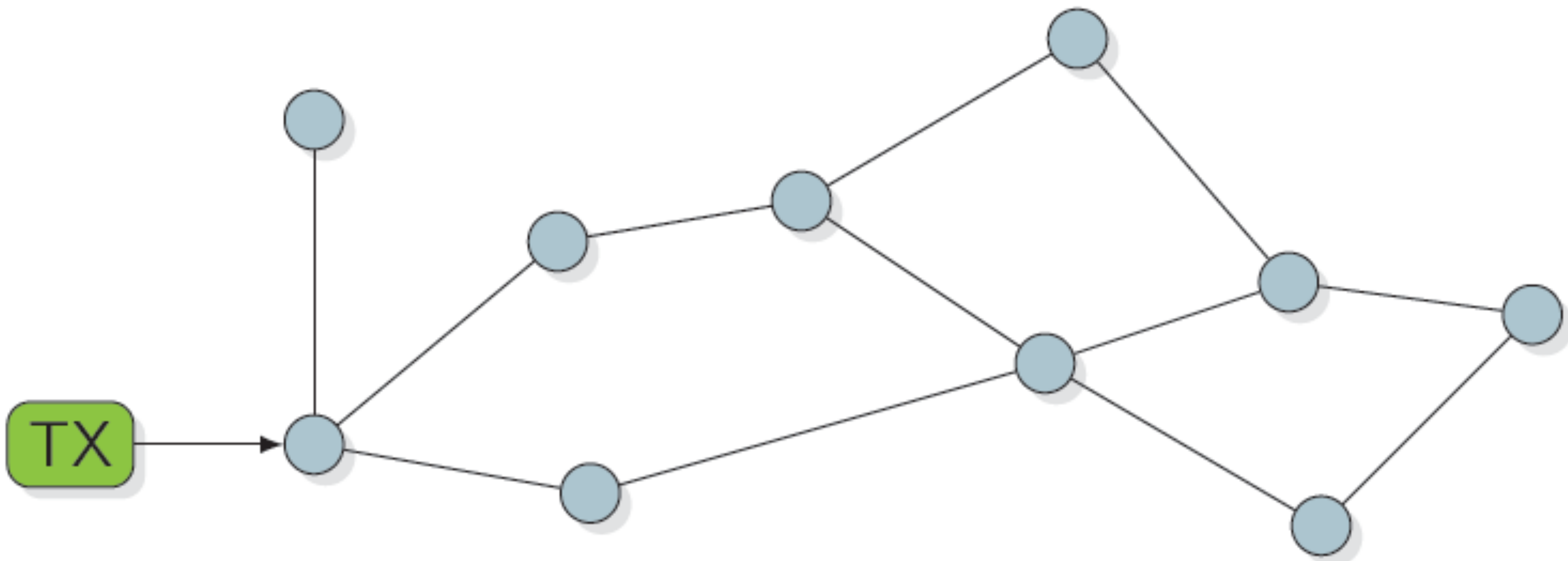
Distributing the Bank



each node has its local ledger

Distributing the Bank

Miner

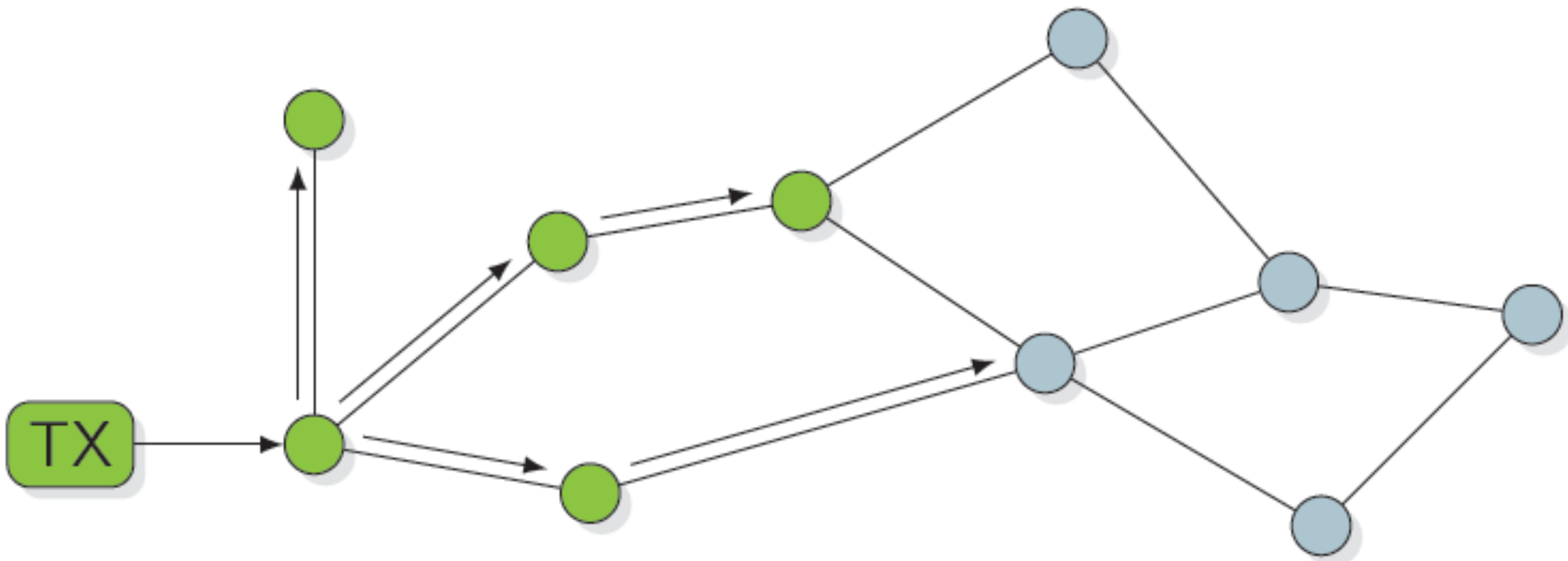


updates its local ledger once receives new transactions

each node has its local ledger

Distributing the Bank

Miner

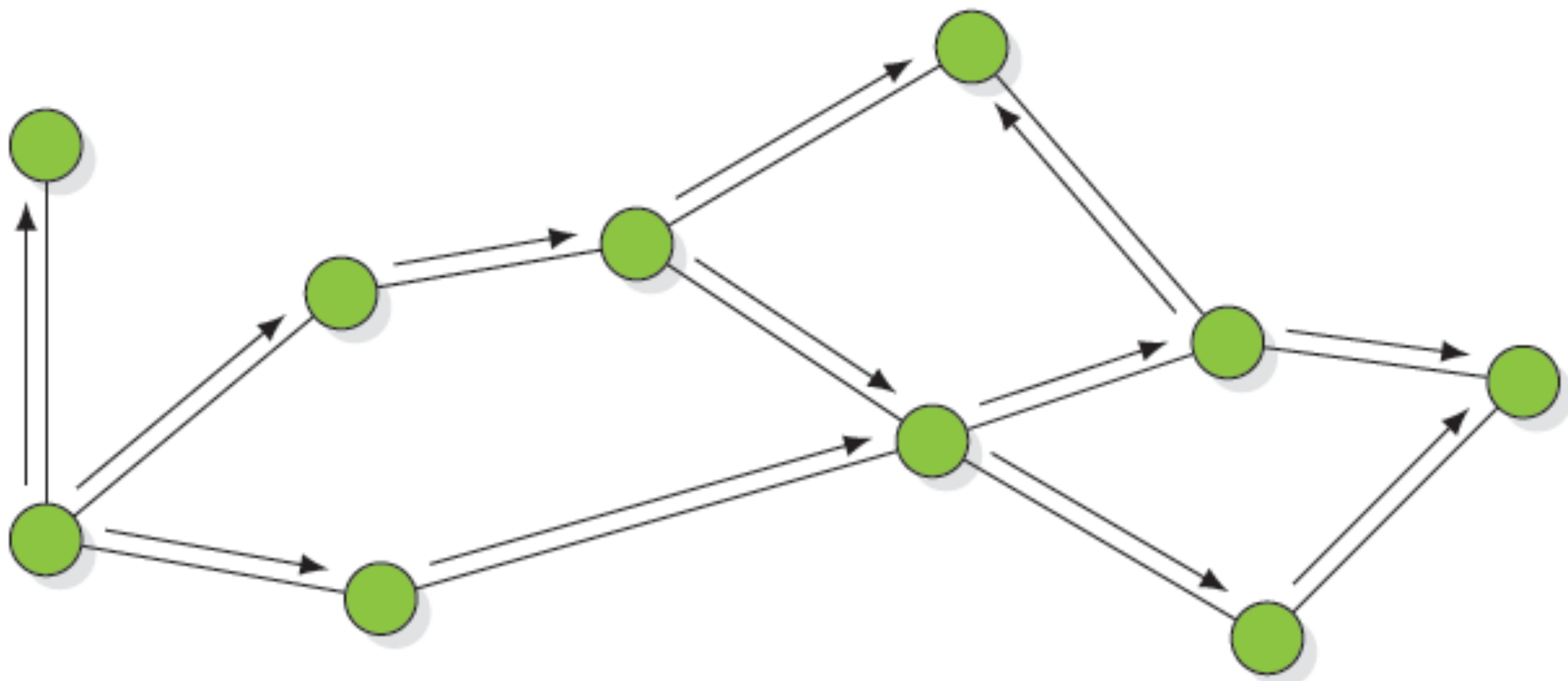


updates its local ledger once receives new transactions

each node has its local ledger

Distributing the Bank

Miner

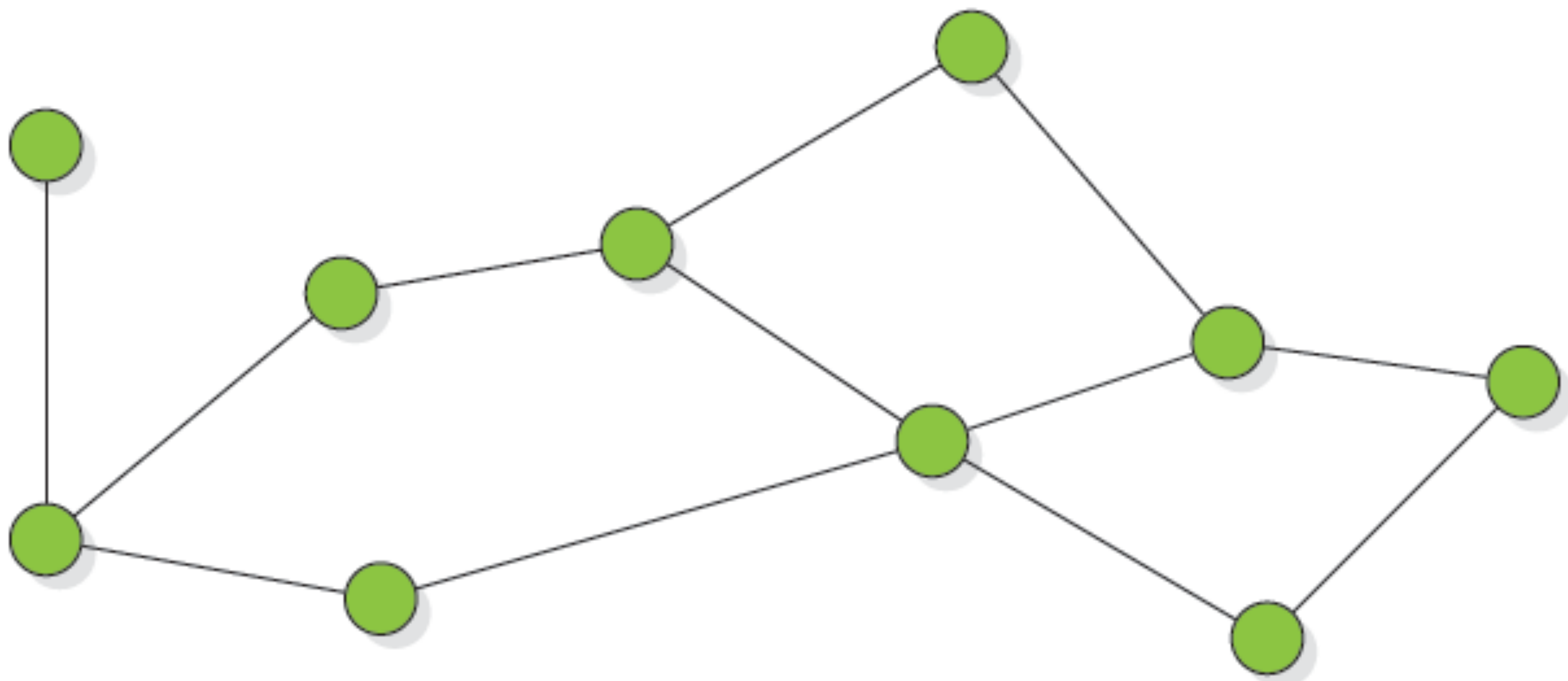


updates its local ledger once receives new transactions

each node has its local ledger

Distributing the Bank

Miner

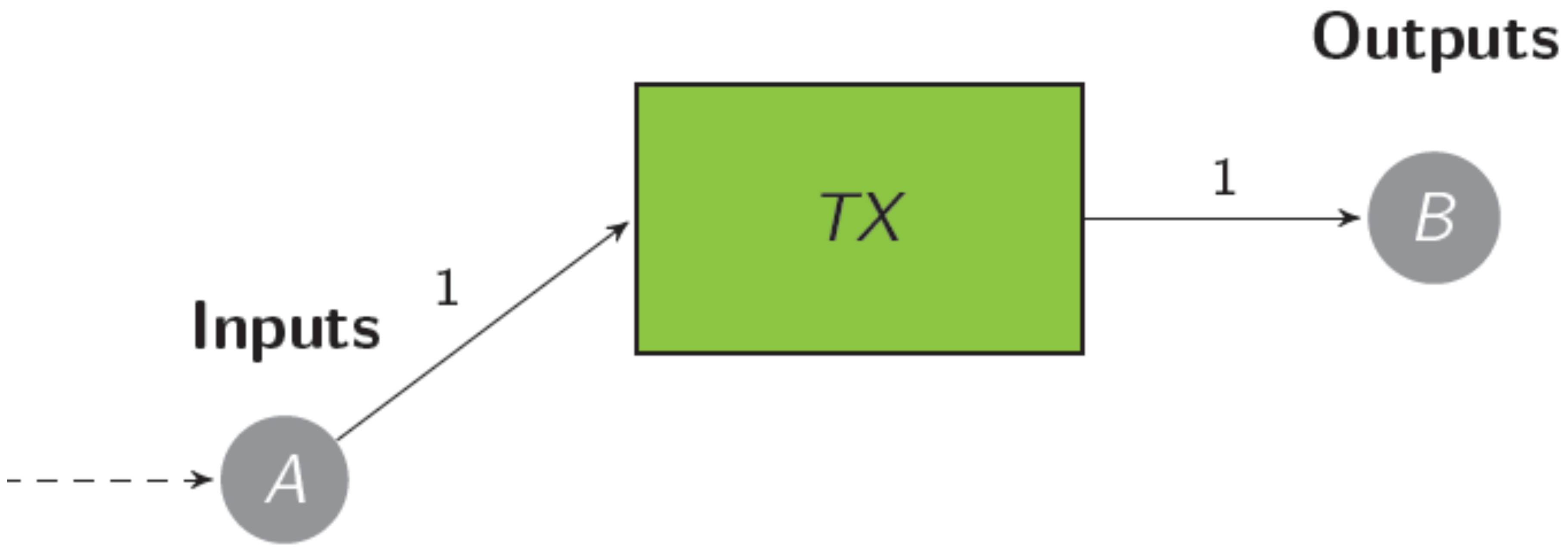


updates its local ledger once receives new transactions

each node has its local ledger

Doublespending

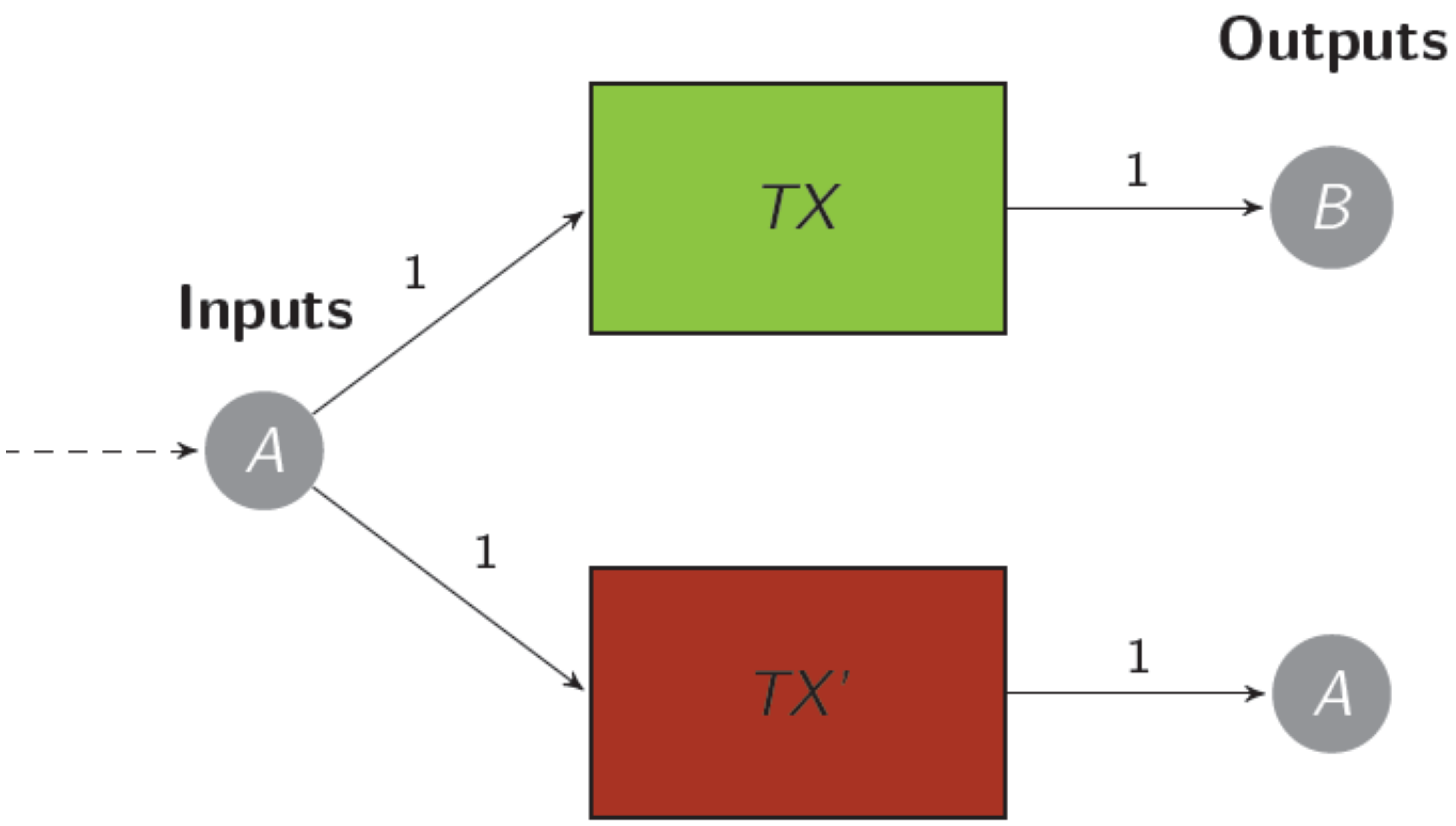
User



conflicted transactions could be generated

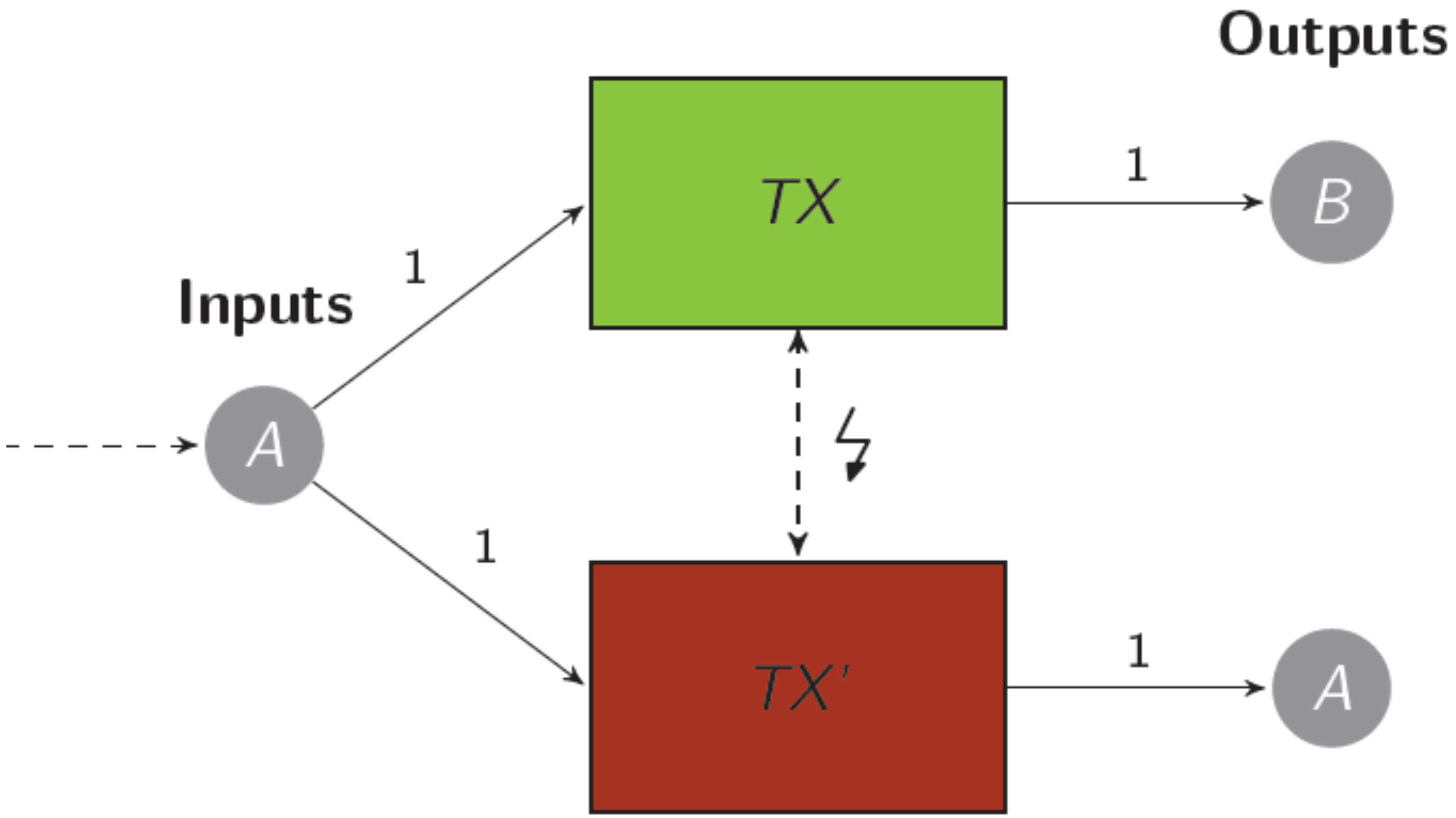
Doublespending

User



conflicted transactions could be generated

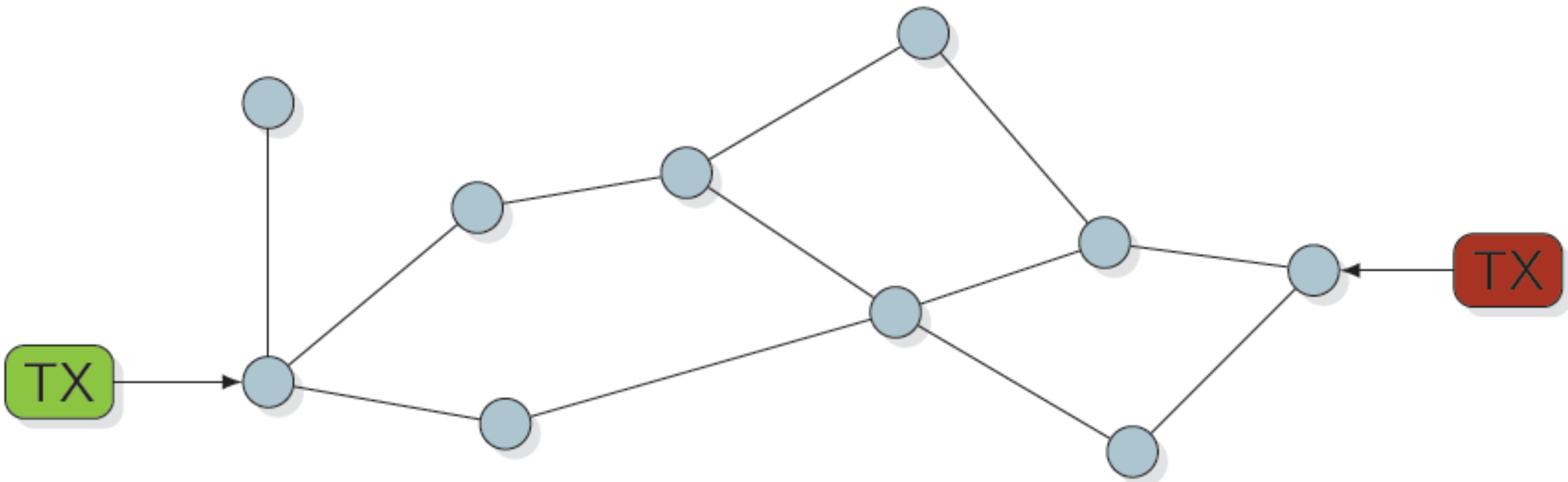
Doublespending



conflicted transactions could be generated

Transaction Conflicts

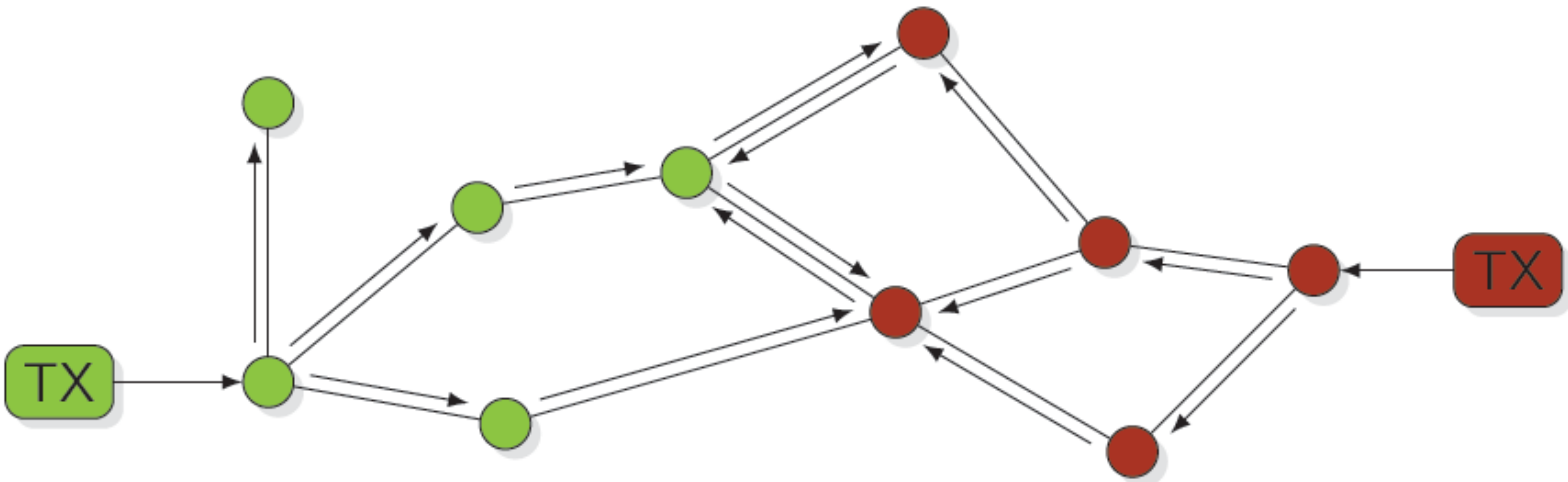
Miner



conflicted transactions appear in the network

Transaction Conflicts

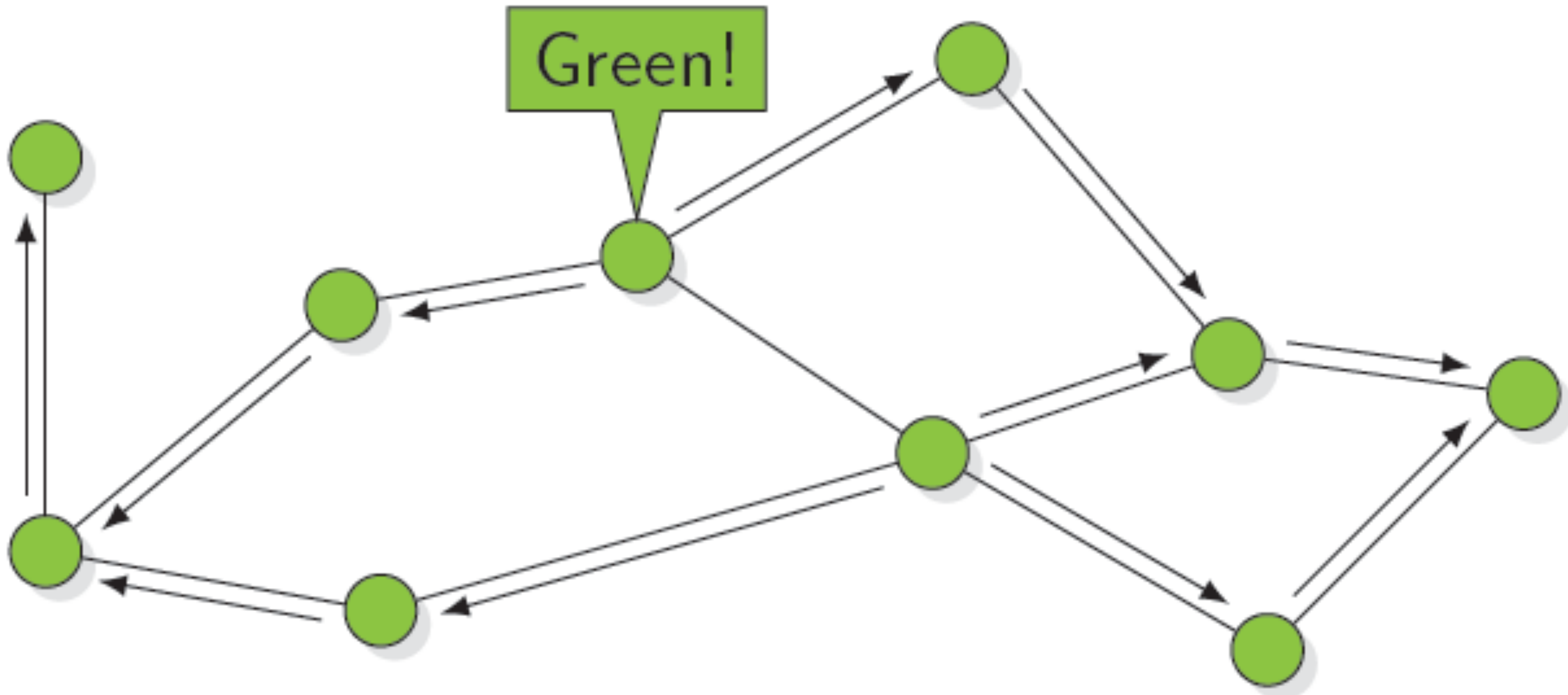
Miner



different nodes may have different local ledgers
conflicted transactions appear in the network

Resolving Conflicts

Miner



to resolve the conflicts, the same ledger must be agreed

How to Choose a Leader?

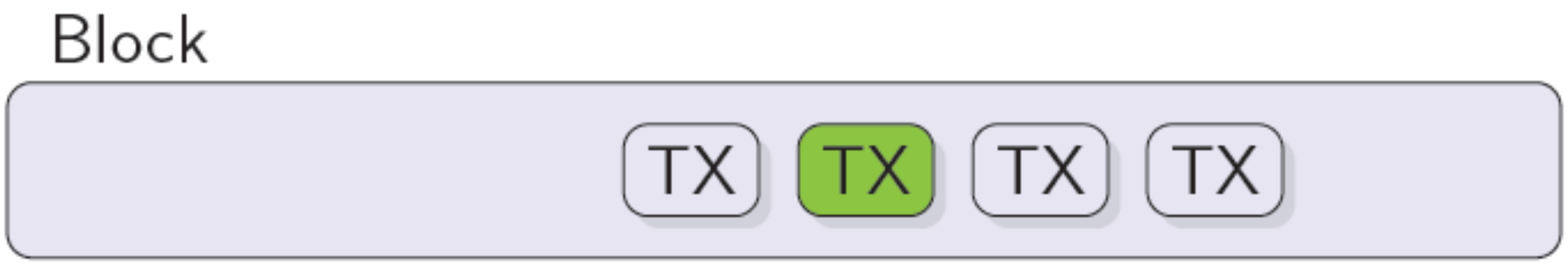
Proof-of-Work

Miner

cryptographic tool:
hash function

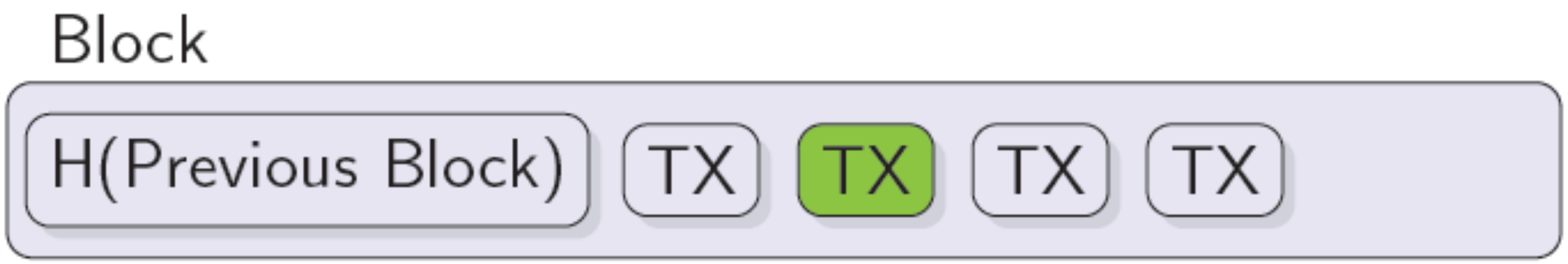
Proof-of-Work

Miner

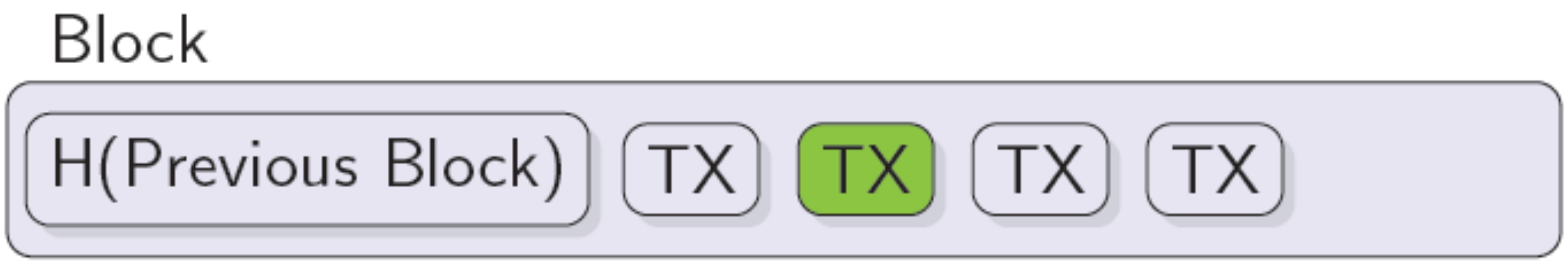


Proof-of-Work

Miner



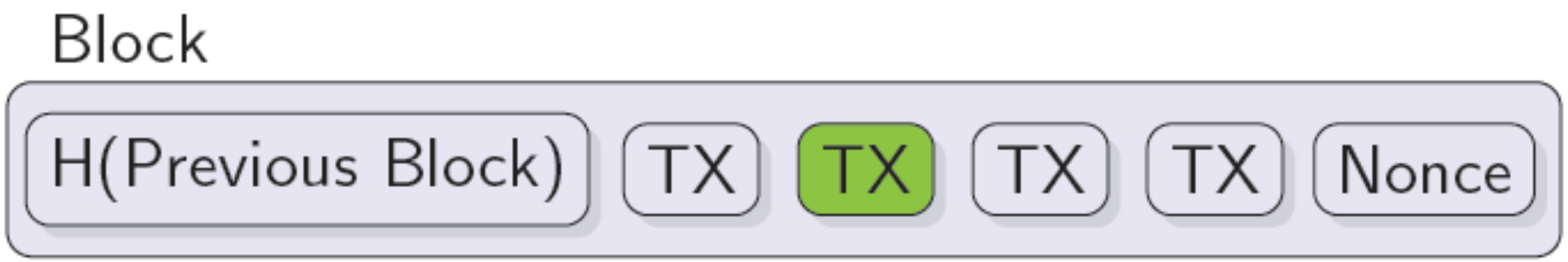
Proof-of-Work



- H(Block) → fd2e2055f117bfa261b5a6c7e11df367...

Proof-of-Work

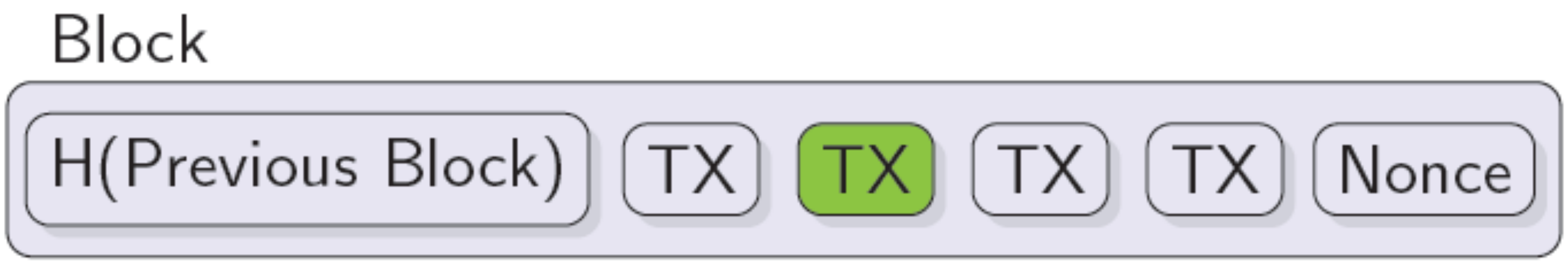
Miner



- $H(\text{Block}|0) \rightarrow$ 094d66aa7c844a9dbb516a41259b5877...

Proof-of-Work

Miner

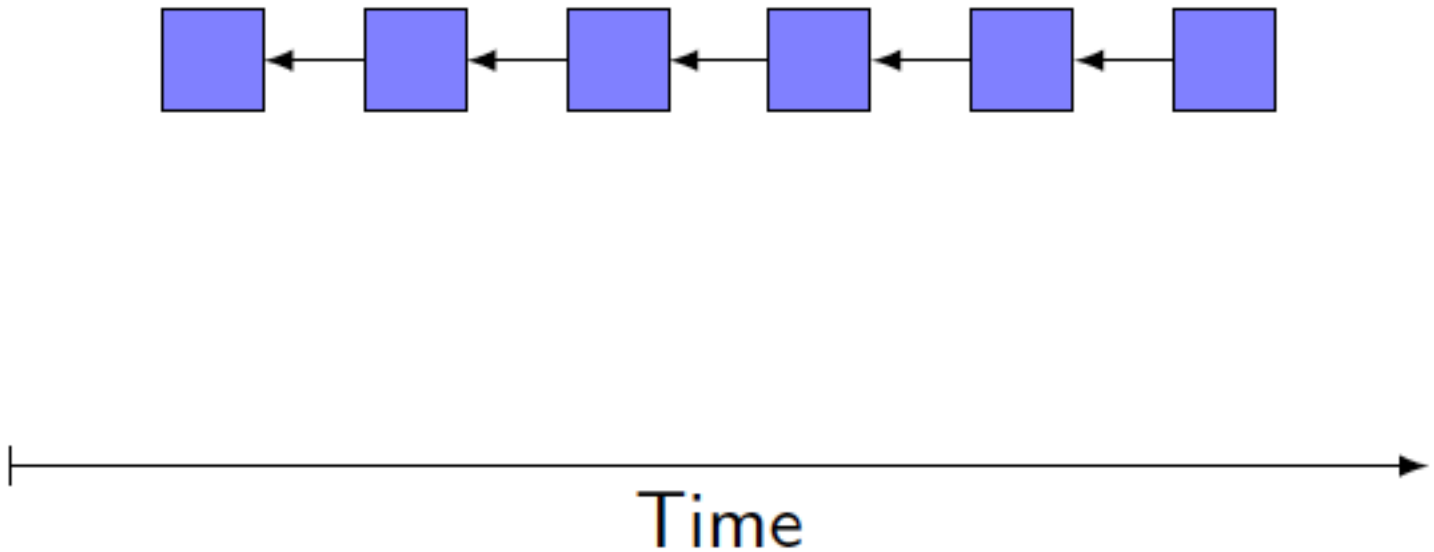


- H(Block|0) $\rightarrow$ 094d66aa7c844a9dbb516a41259b5877...
- H(Block|1) $\rightarrow$ f2496854af8bf989171587a9259f634f...
- H(Block|2) $\rightarrow$ aec87c0ca2e5eb3f23111092f1089ada...
- H(Block|3) $\rightarrow$ 777f75b2a8ecfdc8026c236fc1d2ffa0...

  ⋮

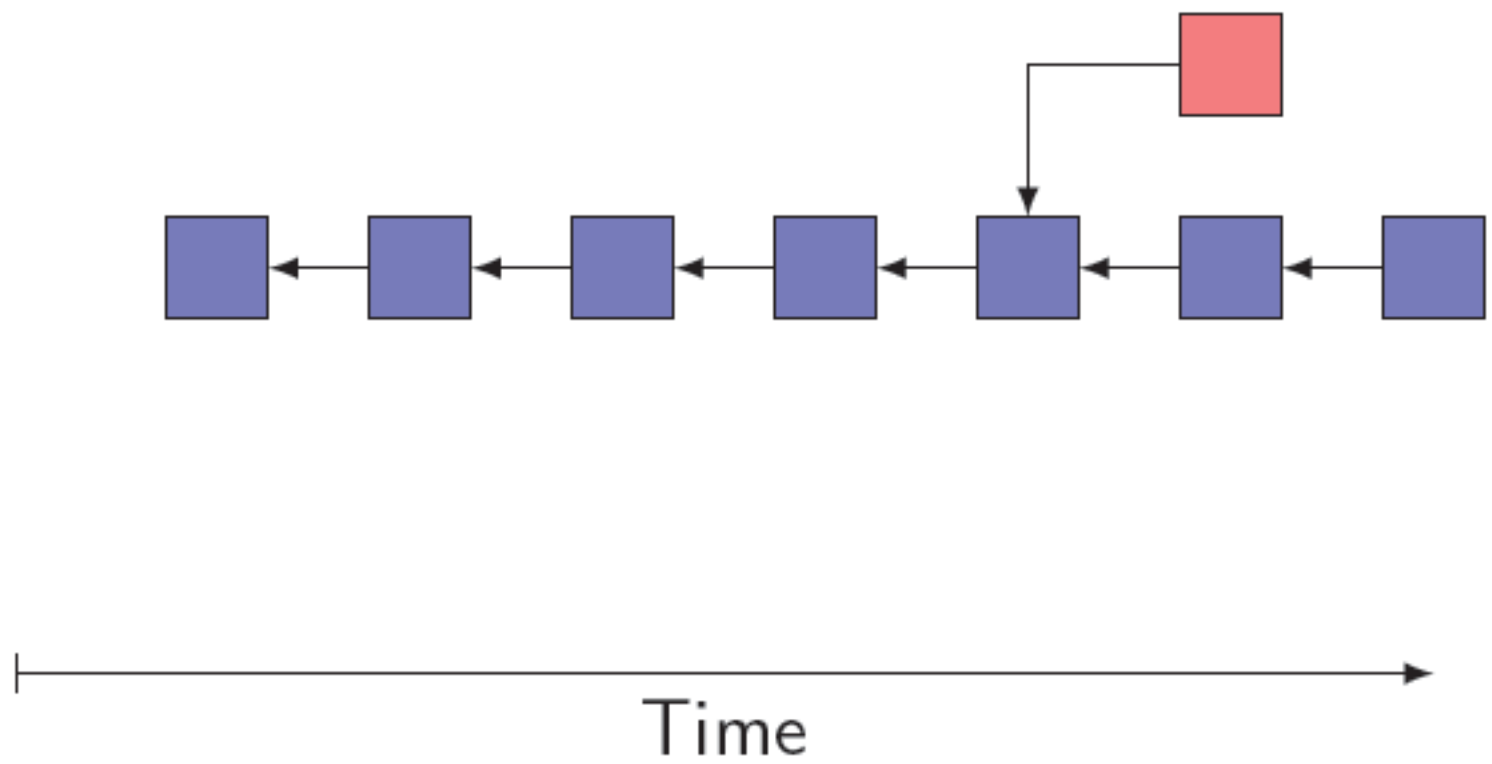
- H(Block|961127) $\rightarrow$ 0000014823419622d4c133672a7d657e...

The Blockchain



The Blockchain

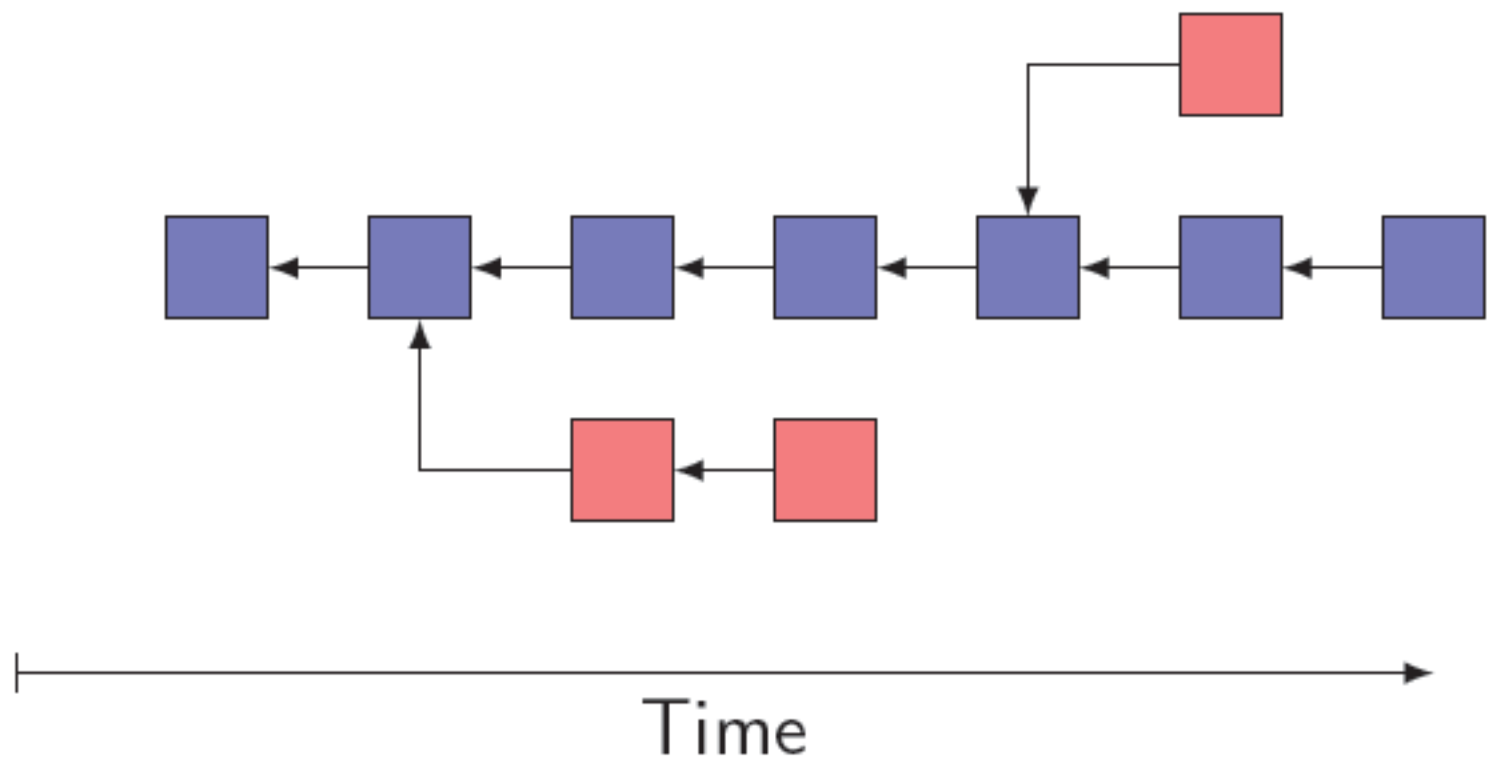
Miner



Is Bitcoin stable?

The Blockchain

Miner



yes, if 51% computing power is from good miners

Is Bitcoin stable?

- two roles: users and **miners**
- distributed protocol

Cryptographic Foundations

A modern approach to building security systems

Crypto Foundations: Why

- Understand the **fundamental security properties** of cryptographic protocols and obtain **proofs of security** in formal adversarial models.

Crypto Foundations: How

- the end goal /objective

- the starting point / building blocks

- the construction: connecting the “starting point” and the “end goal”

- the proof: verifying if the connection is sound

Crypto Foundations: How

- the end goal /objective
  syntax and functionality; security properties

- the starting point / building blocks
  what kinds of resources are available

- the construction: connecting the “starting point” and the “end goal”

- the proof: verifying if the connection is sound

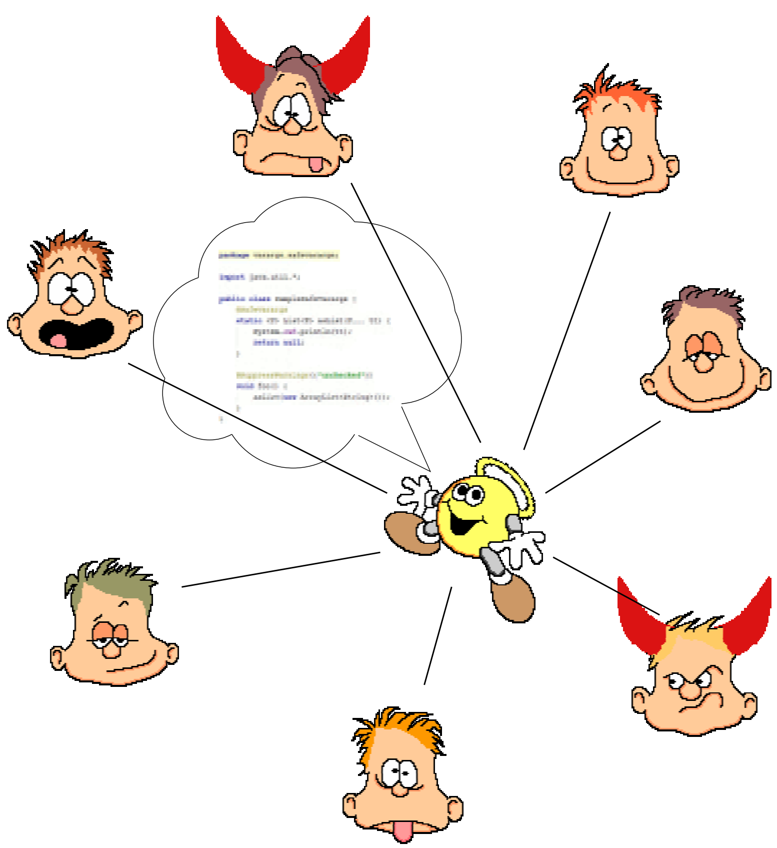
Two popular paradigms

for protocols

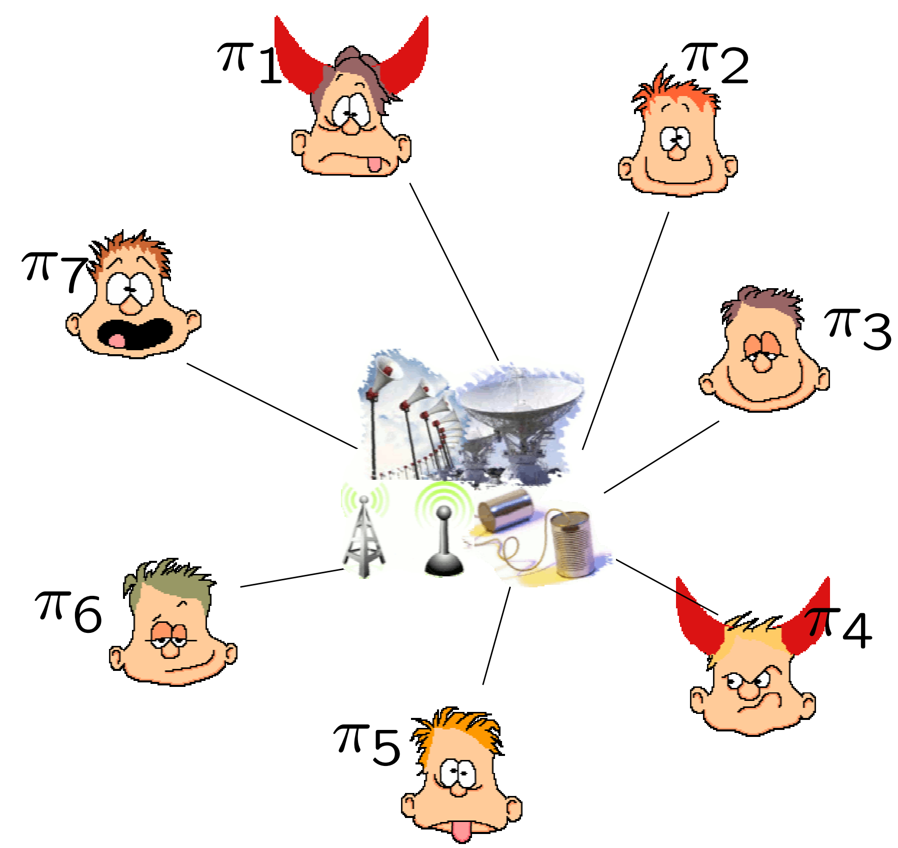
- simulation based
- property based

Simulation based

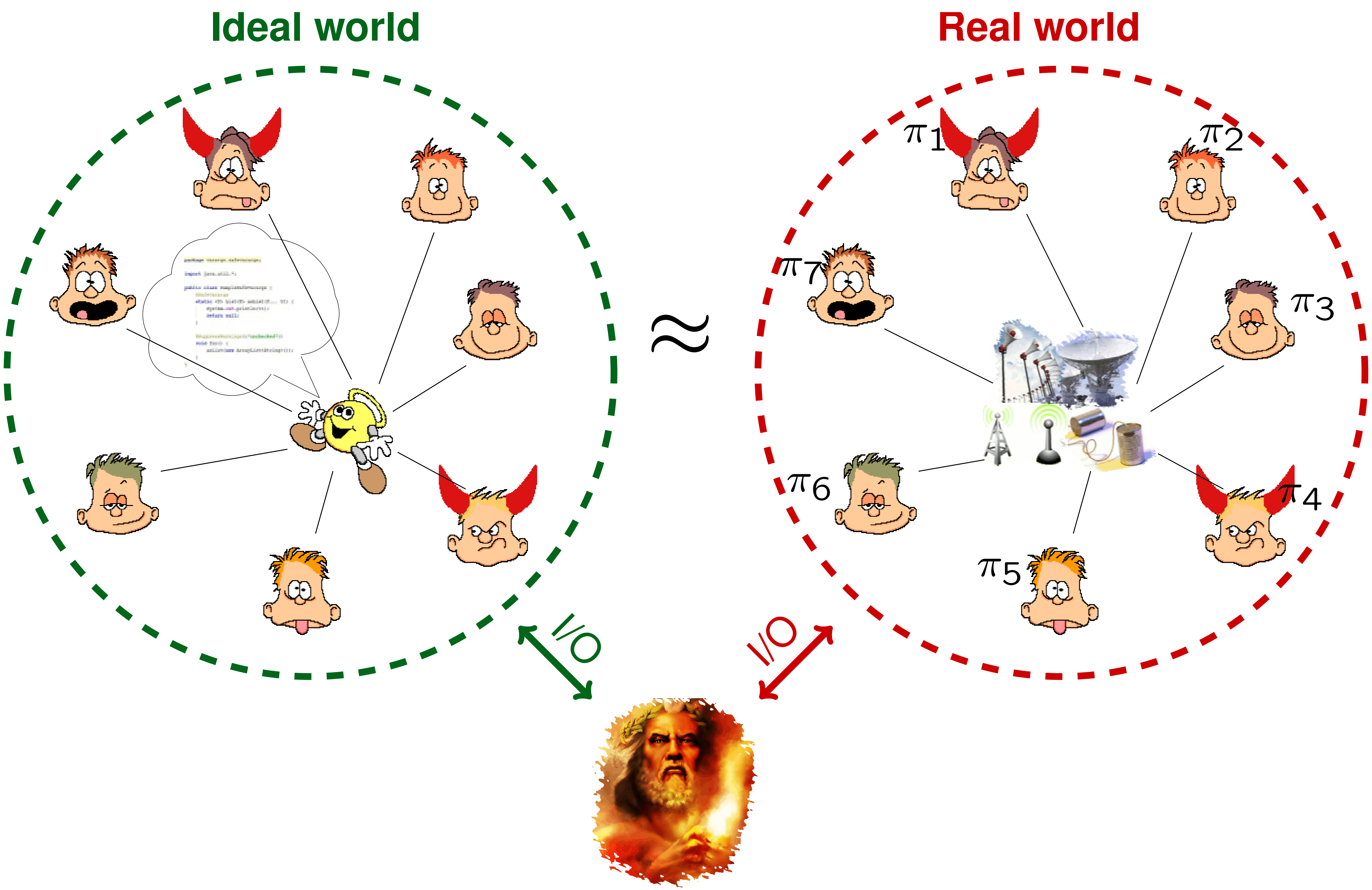
Ideal world | **Real world**



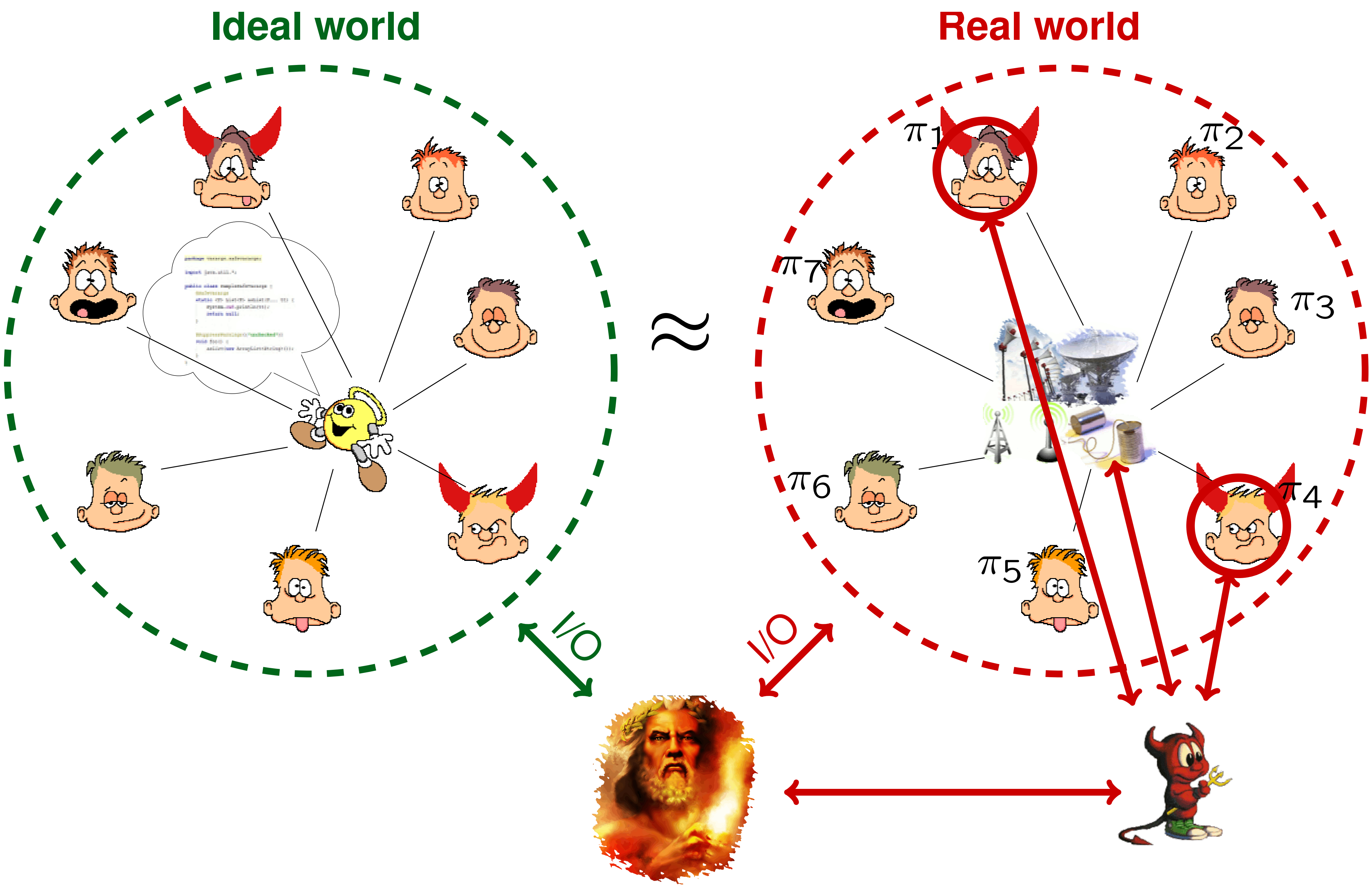
$\approx$



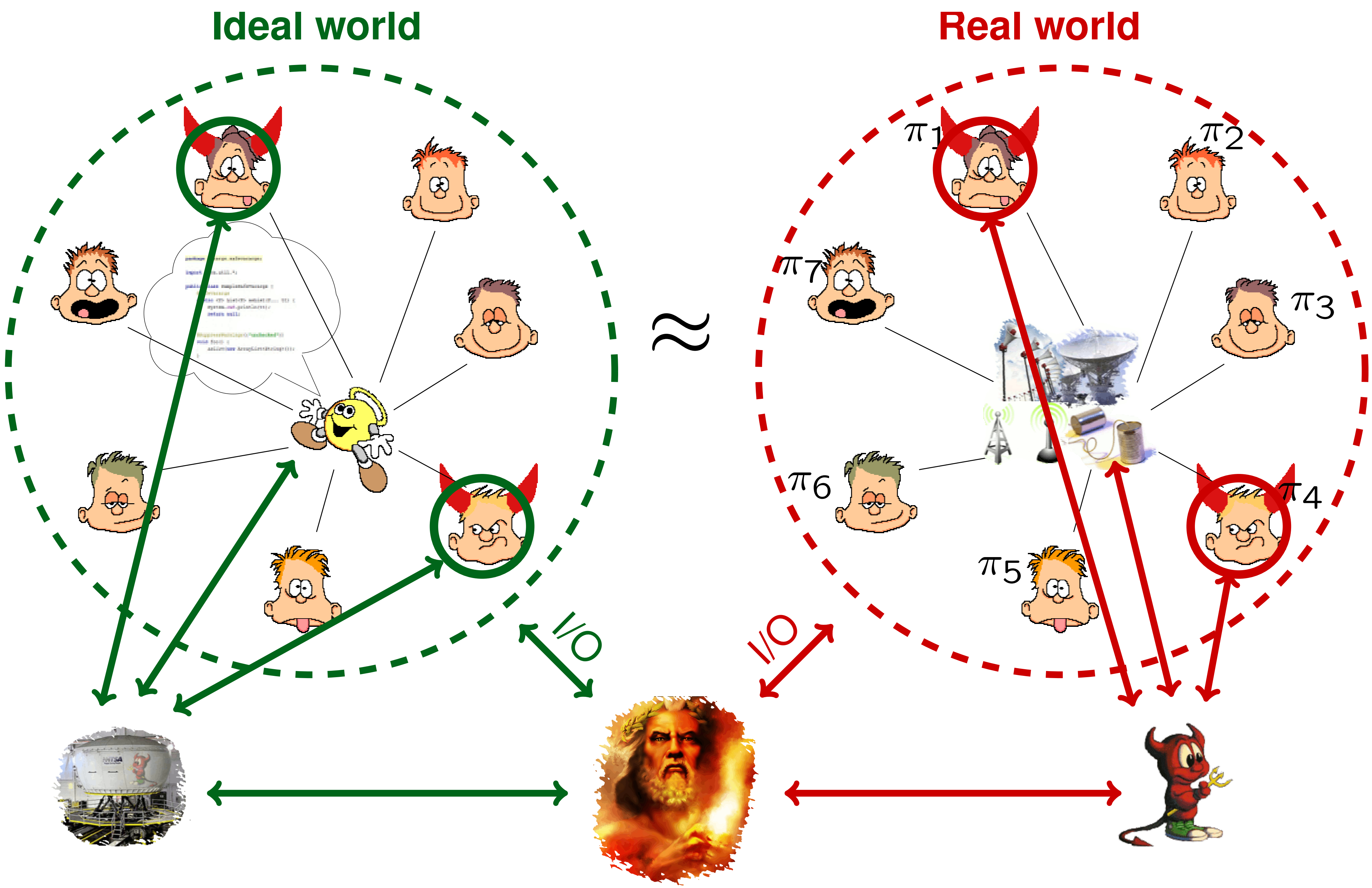
Simulation based



Simulation based



Simulation based



Simulation based

$\forall$  $\exists$  : **Ideal world** $\approx$ **Real world**

…What is an objective?

point to point authenticated communication

Functionality $\mathcal{F}_{\text{AUTH}}$

1. Upon receiving an input $(\texttt{Send}, S, R, sid, m)$ from ITI S, generate a public delayed output $(\texttt{Sent}, S, sid, m)$ to R and halt.
2. Upon receiving $(\texttt{Corrupt-sender}, sid, m')$ from the adversary, and if the $(\texttt{Sent}, S, sid, m)$ output is not yet delivered to R, then output $(\texttt{Sent}, S, sid, m')$ to R and halt.

…What is an objective?

point to point authenticated communication

Functionality $\mathcal{F}_{\text{AUTH}}$

1. Upon receiving an input $(\texttt{Send}, S, R, sid, m)$ from ITI S, generate a public delayed output $(\texttt{Sent}, S, sid, m)$ to R and halt.
2. Upon receiving $(\texttt{Corrupt-sender}, sid, m')$ from the adversary, and if the $(\texttt{Sent}, S, sid, m)$ output is not yet delivered to R, then output $(\texttt{Sent}, S, sid, m')$ to R and halt.

F_{auth} can be realized in the F_{cert} hybrid world

F_{cert} can be realized in the (F_{ca}, F_{sig}) hybrid world

Canetti, *Universally Composable Signature, Certification, and Authentication.* CSF 2004

…What is an objective?

authenticated broadcast

Functionality $\mathcal{F}_{\mathrm{BC}}$

The functionality interacts with an adversary $\mathcal{S}$ and a set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of parties.

- Upon receiving $(\mathsf{Bcast}, \mathsf{sid}, m)$ from P_i, send $(\mathsf{Bcast}, \mathsf{sid}, P_i, m)$ to all parties in $\mathcal{P}$ and to $\mathcal{S}$.

F_{bc} can be realized in the F_{cert} hybrid world

Dolev, Strong, *Authenticated algorithms for Byzantine agreement*. SIAM Journal on Computing, 1983

Hirt, Zikas, *Adaptively Secure Broadcast.* Eurocrypt, 2011

Katz, Garay, Kumaresan, Zhou, *Adaptively Secure Broadcast, Revisited*. PODC, 2011

Simulation based

- Composable; convenient for protocol analysis
- E.g., $\mathcal{F}$ledger based protocol design [Kiayias, **Z**., Zikas 16]

Stay tuned….
my talk tomorrow afternoon about
“Crypto on the blockchain”

Kiayias, Zhou, Zikas, *Fair and Robust Multi-Party Computation using a Global Transaction Ledger.* Eurocrypt 2016

Property based

fix

a protocol Π

a number of parties *n*, *t* of which controlled by adversary

a predicate Q

We say that the protocol has property Q with error ϵ if and only if

$$\forall \mathcal{A}\ \forall \mathcal{Z}\ \text{Prob}[Q(\text{VIEW}^{\Pi}_{\mathcal{A},\mathcal{Z}}(1^{\lambda})] \geq 1 - \epsilon$$

typically : $\epsilon = \text{negl}(\lambda)$

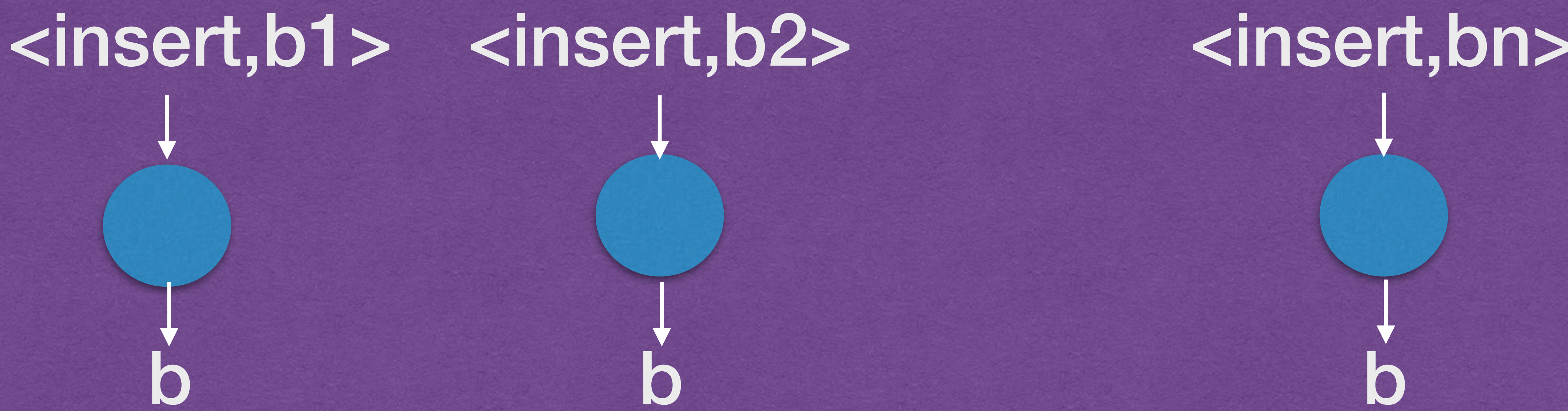
- property based paradigm:
  much restricted adversary/environment;
  advantage: much easier to deal with

Property based
vs
Simulation based

- Simulation based paradigm:
  complex adversary/environment
  advantage: much easier to use

The Consensus Problem

implementing consensus:



Agreement = all parties output the same value

Validity = if all honest parties have the same insert bit, then this matches the output

Termination = all honest parties terminate

Nakamoto's Protocol: The Simplified version

Defining the ledger objective

[Garay, Kiayias, Leonardos 14]

imagine that time is divided in rounds

and protocol organizes transactions in a sequence of blocks

Persistence: parameter k. If an honest party reports a transaction tx as "stable" ($>k$ blocks deep) then, whenever an honest party reports it as stable, it will be in the same position

Liveness: parameters u, k. If all honest parties attempt to insert the transaction *tx* in the ledger, then, after *u rounds*, all honest parties will report it as stable ($>k$ blocks deep) and will always do so

transaction processing time : u as a function of k

Garay, Kiayias, Leonardos, *The Bitcoin Backbone Protocol: Analysis and Applications*. Eurocrypt 2015, IACR ePrint 2014

Synchronous Model

- Time is divided in rounds.
- In each round each party is allowed *q* queries to a hash function (RO)
- messages are sent through a “diffusion” mechanism
- The adversary is rushing and may :
  1. spoof messages
  2. inject messages
  3. reorder messages

Model Participants

- There are (n-t) honest parties each one producing q queries to the hash function per round.
- The adversary is able to control t parties acting as a malicious mining pool.
    - A “flat” version of the world in terms of hashing power.
    - It is worse for honest parties to be separate (they have to pay the price of being decentralized).

Execution & View

3 PPT machines

protocol Π

adversary $\mathcal{A}$

environment $\mathcal{Z}$

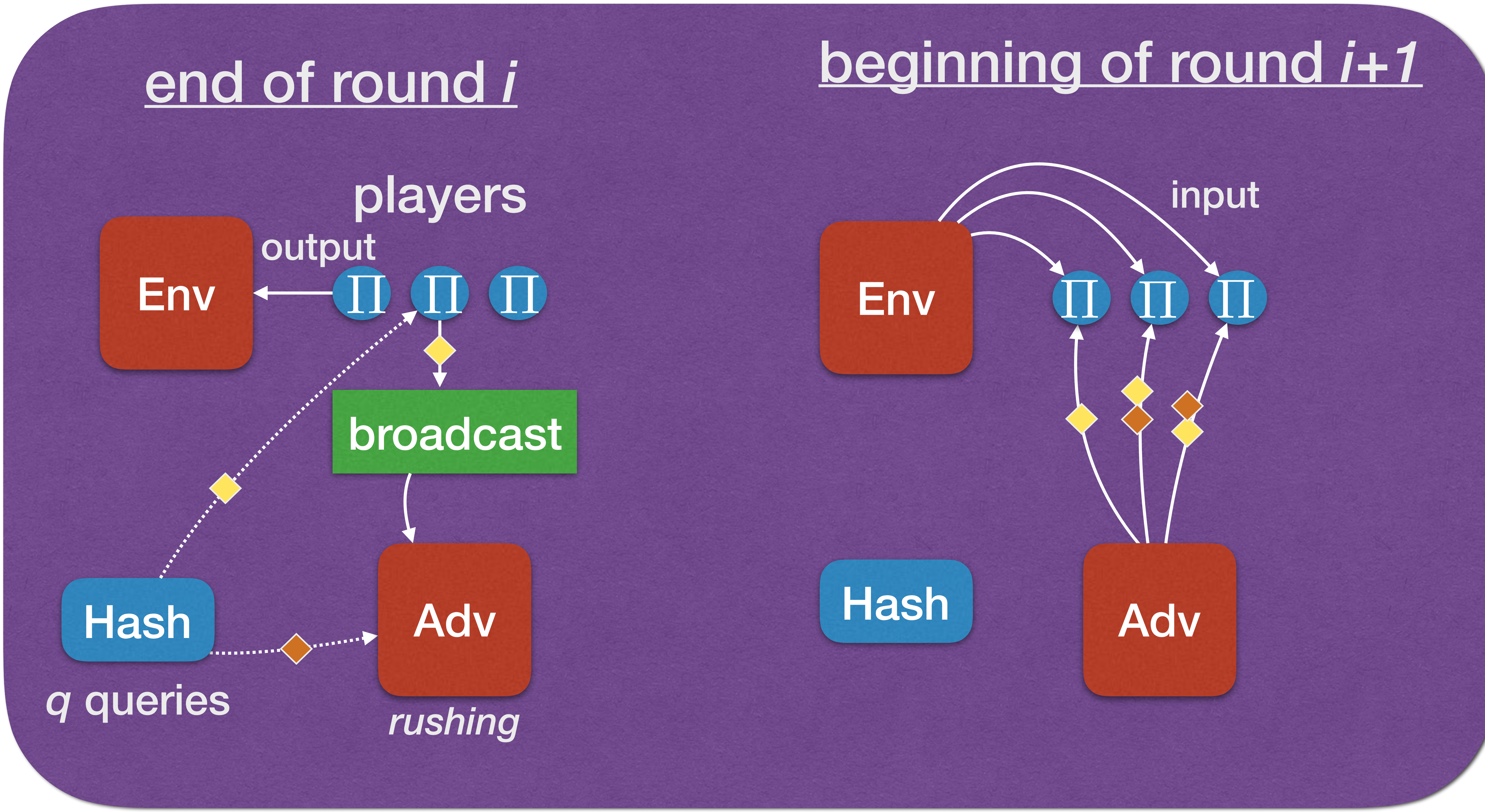
n parties

$\text{VIEW}^{\Pi}_{\mathcal{A},\mathcal{Z}}(1^{\lambda})$ concatenation of the view of each party at each round

random variable with support :

1. coins of $\mathcal{A}, \mathcal{Z}, n$ copies of Π
2. Random oracle

Round structure



Recall: Property of a protocol

fix

a protocol Π

a number of parties *n*, *t* of which controlled by adversary

a predicate Q

We say that the protocol has property Q with error ϵ if and only if

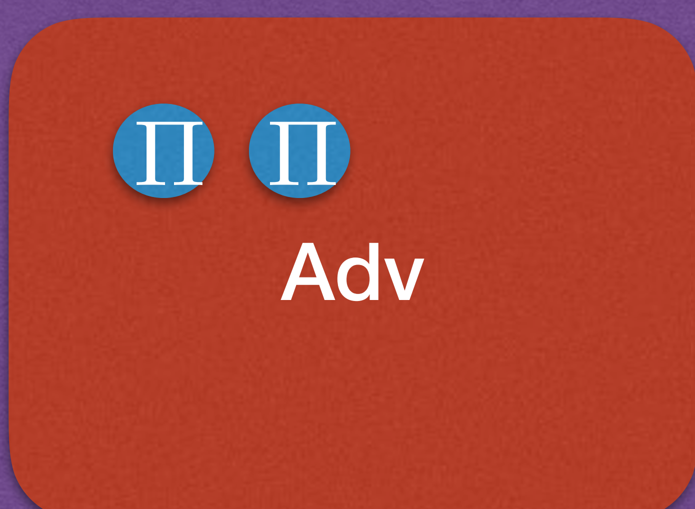
$$\forall \mathcal{A}\ \forall \mathcal{Z}\ \mathsf{Prob}[Q(\mathsf{VIEW}^{\Pi}_{\mathcal{A},\mathcal{Z}}(1^{\lambda})] \geq 1 - \epsilon$$

typically : $\epsilon = \mathsf{negl}(\lambda)$

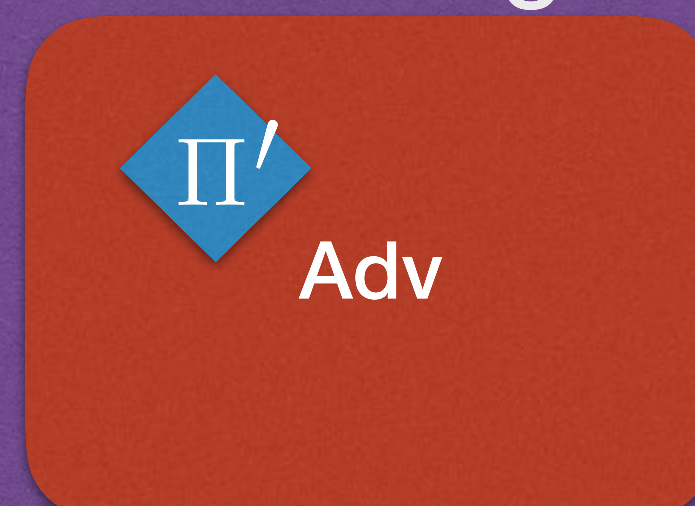
Generality of the model

- We quantify over all possible adversaries; this includes:

some parties receiving only some of the messages



a large mining pool that is performing some type of selfish mining



<u>Or any combination thereof</u>

Nakamoto's Protocol: The Simplified version

Backbone protocol

The Bitcoin Backbone Protocol

[Garay, Kiayias, Leonardos 14]

- An abstraction based on the Bitcoin implementation.
  - **Importantly** : it distinguishes between data structure (blockchain) and application layer (transactions).

Bitcoin Backbone (1)

parameterized by $V(\cdot), I(\cdot), R(\cdot)$
and $G(\cdot), H(\cdot)$ hash functions

- players have a state $\mathcal{C}$ in the form of a "blockchain":

The *contents* of $\mathcal{C}$ satisfy the predicate

$$V(x_1, \ldots, x_i) = \text{true}$$

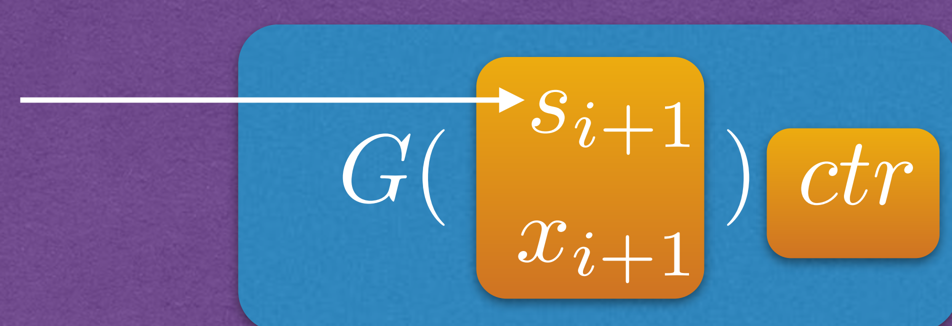
Bitcoin Backbone (2)

parameterized by $V(\cdot), I(\cdot), R(\cdot)$
and $G(\cdot), H(\cdot)$ hash functions

- Within a round, players obtain (INSERT, x) symbols from the environment and network and process them

$$x_{i+1} = I(\ldots \text{all local info} \ldots)$$

- Then they use their *q* queries to $H(\cdot)$ to obtain a new block by trying $ctr = 0, 1, 2, \ldots$



Bitcoin Backbone (3)

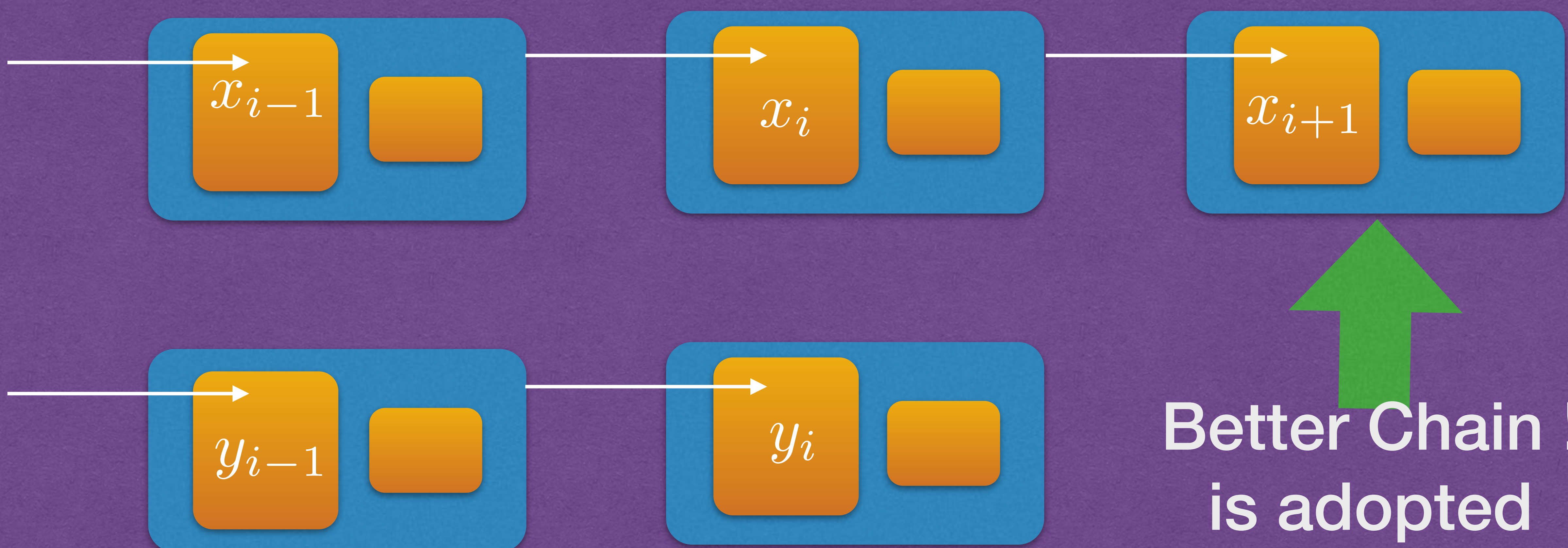
parameterized by $V(\cdot), R(\cdot), I(\cdot)$

- If a player finds a new block it extends $\mathcal{C}$

- The new $\mathcal{C}$ is propagated to all players via the (unreliable/anonymous) broadcast

Bitcoin Backbone (4)

- A player will compare any incoming chains and the local chain w.r.t. their length/difficulty



- Finally a player given a (Read) symbol it will return

$$R(x_1, x_2, \ldots, x_{i+1})$$

Validate

function validate($\mathcal{C}$)
  $b \leftarrow V(\mathbf{x}_{\mathcal{C}})$
  if $b \wedge (\mathcal{C} \neq \varepsilon)$ **then** ▷ The chain is non-empty and meaningful w.r.t. $V(\cdot)$
    $\langle s, x, ctr \rangle \leftarrow \text{head}(\mathcal{C})$
    $s' \leftarrow H(ctr, G(s, x))$
    repeat
      $\langle s, x, ctr \rangle \leftarrow \text{head}(\mathcal{C})$
      if $\text{validblock}_q^T(\langle s, x, ctr \rangle) \wedge (H(ctr, G(s, x)) = s')$ **then**
        $s' \leftarrow s$ ▷ Retain hash value
        $\mathcal{C} \leftarrow \mathcal{C}^{\lceil 1}$ ▷ Remove the head from $\mathcal{C}$
      else
        $b \leftarrow \text{False}$
      end if
    until $(\mathcal{C} = \varepsilon) \vee (b = \text{False})$
  end if
  return (b)
end function

POW

```
function pow(x, C)
    if C = ε then                                   ▷ Determine proof of work instance
        s ← 0
    else
        ⟨s′, x′, ctr′⟩ ← head(C)
        s ← H(ctr′, G(s′, x′))
    end if
    ctr ← 1
    B ← ε
    h ← G(s, x)
    while (ctr ≤ q) do
        if (H(ctr, h) < T) then                     ▷ This H(·) invocation subject to the q-bound
            B ← ⟨s, x, ctr⟩
            break
        end if
        ctr ← ctr + 1
    end while
    C ← CB                                          ▷ Extend chain
    return C
end function
```

Main Loop

$\mathcal{C} \leftarrow \varepsilon$
$state \leftarrow \varepsilon$
$round \leftarrow 0$
while TRUE **do**
    $\widetilde{\mathcal{C}} \leftarrow \mathsf{maxvalid}(\mathcal{C}$, all chains found in RECEIVE())
    $\langle state, x \rangle \leftarrow I(state, \widetilde{\mathcal{C}}, round,$ INPUT(), RECEIVE()) ▷ Determine the x-value.
    $\mathcal{C}_{\mathsf{new}} \leftarrow \mathsf{pow}(x, \widetilde{\mathcal{C}})$
    if $\mathcal{C} \neq \mathcal{C}_{\mathsf{new}}$ **then**
        $\mathcal{C} \leftarrow \mathcal{C}_{\mathsf{new}}$
        BROADCAST($\mathcal{C}$)
    end if
    $round \leftarrow round + 1$
    if INPUT() contains READ **then**
        write $R(\mathbf{x}_{\mathcal{C}})$ to OUTPUT()
    end if
end while

Warning
broadcasting the chain C is not OK in reality

Nakamoto's Protocol: The Simplified version

Blockchain properties

Backbone Protocol Properties

Common Prefix

(informally)

If two players prune a sufficient number of blocks from their chains they will obtain the same prefix

Chain Quality

(informally)

Any (large enough) chunk of an honest player's chain will contain some blocks from honest players

Chain Growth

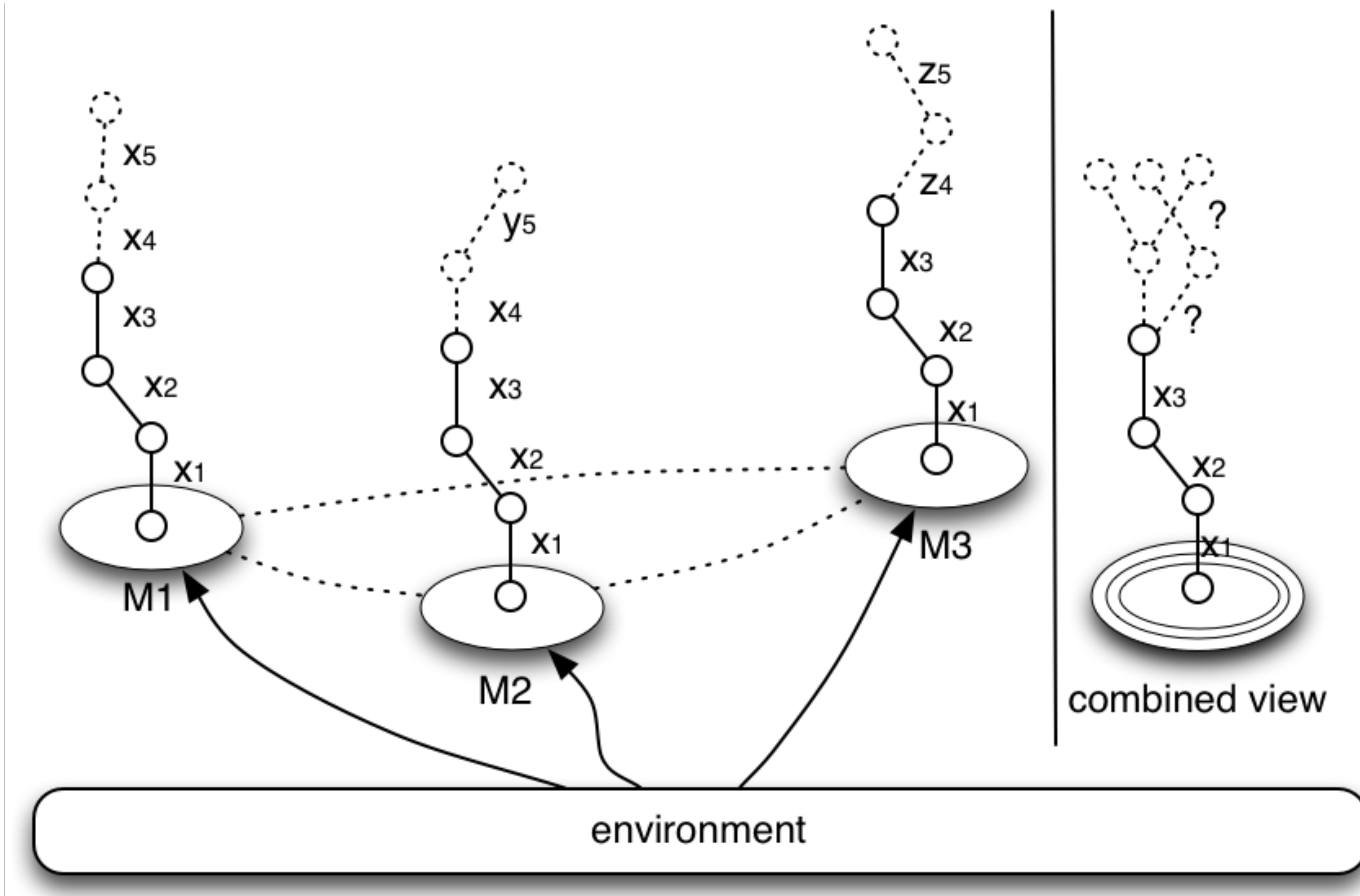
(informally)

the chain of any honest player grows at least at a steady rate - the chain speed coefficient

Based on work of [GKL14, KP15]

CP: will honest players converge?

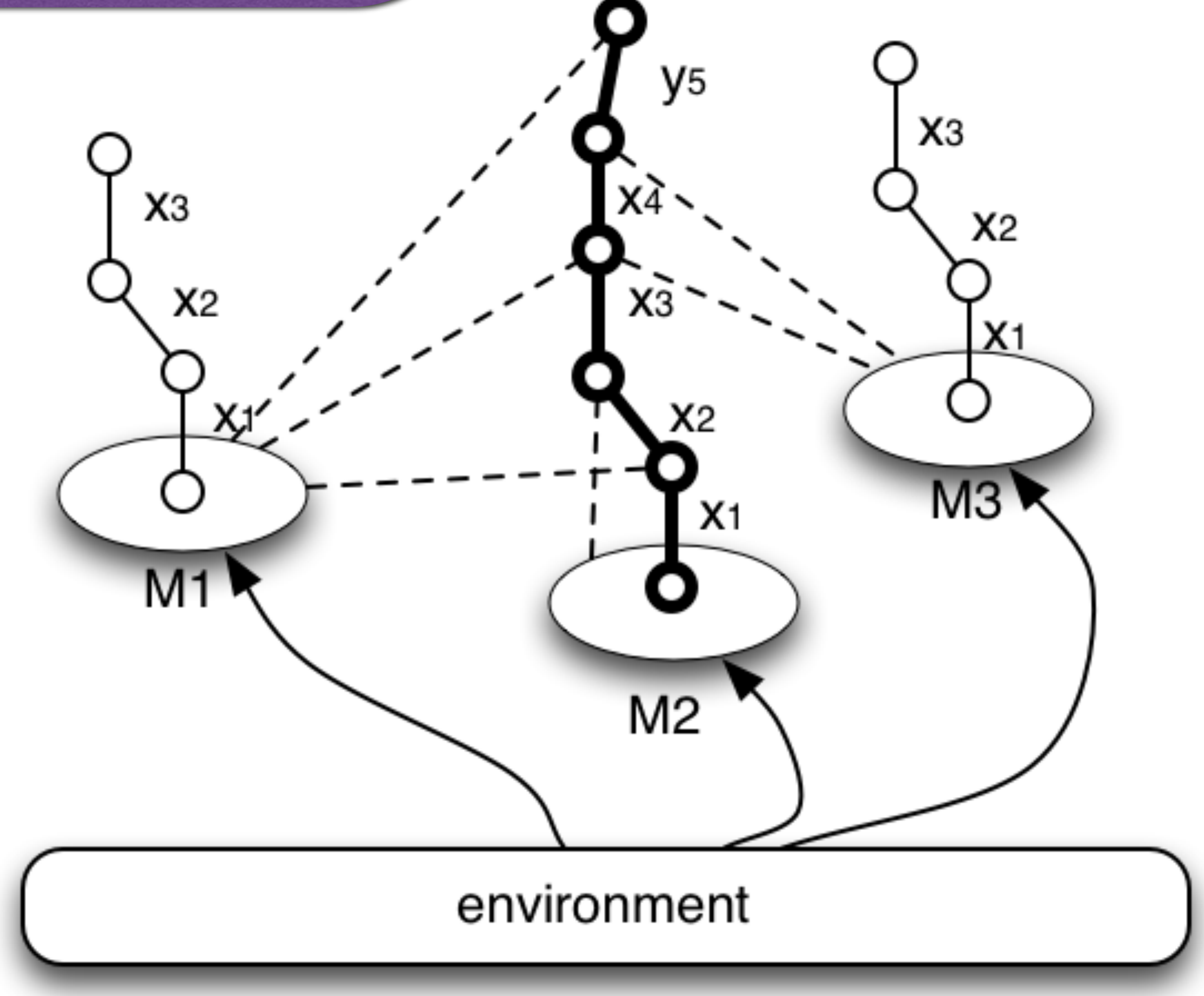
$\forall r_1, r_2, (r_1 \leq r_2), P_1, P_2,$ with $\mathcal{C}_1, \mathcal{C}_2 : \ \mathcal{C}_1^{\lceil k} \preceq \mathcal{C}_2$



CQ: are honest blocks going to be adopted by the parties?

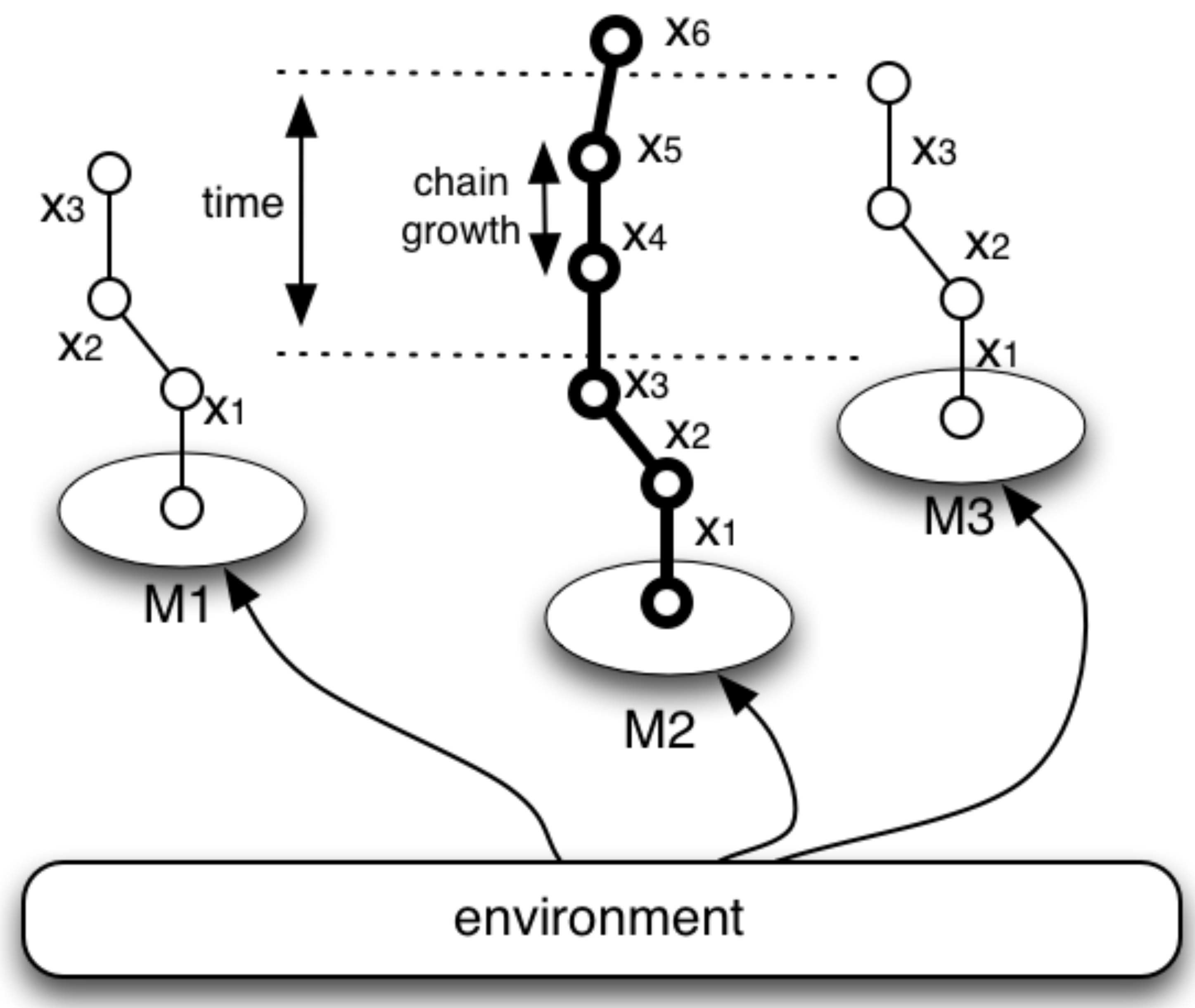
Parameters $\mu \in (0,1), k \in \mathbb{N}$

The proportion of blocks in any k-long subsequence produced by the adversary is less than μk



Chain Growth: does the chain grow?

Parameters $\tau \in (0,1), s \in \mathbb{N}$
$\forall r_1, r_2$ honest player P with chains $\mathcal{C}_1, \mathcal{C}_2$
$r_2 - r_1 \geq s \implies |\mathcal{C}_2| - |\mathcal{C}_1| \geq \tau s$



Proof strategy

1. Define the notion of *typical execution.*

2. Argue that typical executions have with overwhelming probability.

3. Prove CG, CP, CQ

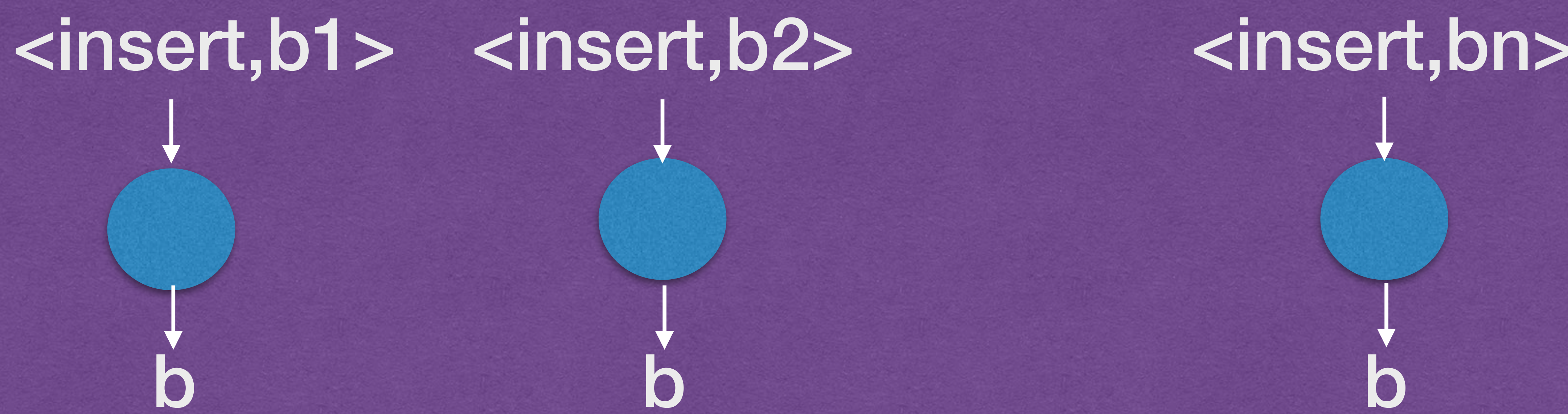
4. Derive persistence and liveness.

Nakamoto's Protocol: The Simplified version

Applications

Recall : Consensus

implementing consensus:



Agreement = all parties output the same value

Validity = if all honest parties have the same insert bit, then this matches the output

Termination = all honest parties terminate

Applying the backbone protocol

- It is all about defining V, I, R :
  - V = validity predicate.
  - I = input function
  - R = read function

The Nakamoto "consensus protocol"

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto | Thu, 13 Nov 2008 19:34:25 -0800

```
James A. Donald wrote:
> It is not sufficient that everyone knows X. We also
> need everyone to know that everyone knows X, and that
> everyone knows that everyone knows that everyone knows X
> - which, as in the Byzantine Generals problem, is the
> classic hard problem of distributed data processing.

The proof-of-work chain is a solution to the Byzantine Generals' Problem.  I'll
try to rephrase it in that context.

A number of Byzantine Generals each have a computer and want to attack the
King's wi-fi by brute forcing the password, which they've learned is a certain
number of characters in length.  Once they stimulate the network to generate a
packet, they must crack the password within a limited time to break in and
erase the logs, otherwise they will be discovered and get in trouble.  They
only have enough CPU power to crack it fast enough if a majority of them attack
at the same time.

They don't particularly care when the attack will be, just that they all agree.
 It has been decided that anyone who feels like it will announce a time, and
whatever time is heard first will be the official attack time.  The problem is
```

https://www.mail-archive.com/cryptography@metzdowd.com/msg09997.html

Applying the backbone for consensus, (1)

Nakamoto "consensus protocol"

Content validation predicate $V(\cdot)$	$V(\langle x_1, \ldots, x_n \rangle)$ is true if and only if it holds that $v_1 = \ldots = v_n \in \{0,1\}, \rho_1, \ldots, \rho_n \in \{0,1\}^\kappa$ where $x_i = \langle v_i, \rho_i \rangle$.
Chain reading function $R(\cdot)$ (parameterized by k)	If $V(x_\mathcal{C})$ = True and $\text{len}(\mathcal{C}) \geq k$, the value of $R(\mathcal{C})$ is the (unique) value v that is present in each block of $\mathcal{C}$, while it is undefined if $V(x_\mathcal{C})$ = False or $\text{len}(\mathcal{C}) < k$.
Input contribution function $I(\cdot)$	If $\mathcal{C} = \emptyset$ and (INSERT, v) is in the input tape then $I(st, \mathcal{C}, round, \text{INPUT}())$ is equal to $\langle v, \rho \rangle$ where $\rho \in \{0,1\}^\kappa$ is a random value; otherwise (i.e., the case $\mathcal{C} \neq \emptyset$), it is equal to $\langle v, \rho \rangle$ where v is the unique $v \in \{0,1\}$ value that is present in $\mathcal{C}$ and $\rho \in \{0,1\}^\kappa$ is a random value. The state st always remains ϵ.

It works .. but only with constant probability of success (not overwhelming)

Applying the backbone for consensus, (2)

A (1/3) "consensus protocol" (from GKL14)

Content validation predicate $V(\cdot)$	$V(\langle x_1, \ldots, x_n \rangle)$ is true if and only if $v_1, \ldots, v_n \in \{0,1\}, \rho_1, \ldots, \rho_n \in \{0,1\}^\kappa$ where v_i, ρ_i are the values from the pair $x_i = \langle v_i, \rho_i \rangle$.
Chain reading function $R(\cdot)$ (parameterized by k)	If $V(\langle x_1, \ldots, x_n \rangle) = \text{True}$ and $n \geq 2k$, the value $R(\mathcal{C})$ is the majority bit of $v_1, \ldots, v_k$ where $x_i = \langle v_i, \rho_i \rangle$; otherwise (i.e., the case $V(\langle x_1, \ldots, x_n \rangle) = \text{False}$ or $n < 2k$) the output value is undefined.
Input contribution function $I(\cdot)$	$I(st, \mathcal{C}, round, \text{INPUT}())$ is equal to $\langle v, \rho \rangle$ if the input tape contains (INSERT, v); ρ is a random κ-bit string. The state st remains always ϵ.

It works .. but only up to 1/3 adversarial power.

Applying the backbone for transaction ledger

(from GKL14)

Content validation predicate $V(\cdot)$	$V(\langle x_1, \ldots, x_m \rangle)$ is true if and only if the vector $\langle x_1, \ldots, x_m \rangle$ is a valid ledger, i.e., $\langle x_1, \ldots, x_m \rangle \in \mathcal{L}$.
Chain reading function $R(\cdot)$	If $V(\langle x_1, \ldots, x_m \rangle) = \text{True}$, the value $R(\mathcal{C})$ is equal to $\langle x_1, \ldots, x_m \rangle$; undefined otherwise.
Input contribution function $I(\cdot)$	$I(st, \mathcal{C}, round, \textsc{Input}())$ operates as follows: if the input tape contains $(\textsc{Insert}, v)$, it parses v as a sequence of transactions and retains the largest subsequence $x' \preceq v$ that is valid with respect to $\mathbf{x}_{\mathcal{C}}$ (and whose transactions are not already included in $\mathbf{x}_{\mathcal{C}}$). Finally, $x = \text{tx}_0 x'$ where tx_0 is a neutral random nonce transaction.

it satisfies persistence and liveness with overwhelming probability as long as also digital signature security holds.

Nakamoto's Protocol: The Full-fledged version

Can we have a complete analysis ?

several progresses for analyzing the simplified Nakamoto protocol

- Pass et al, Eurocrypt 17, more realistic network
- Garay et al, Crypto17, adaptive difficulty adjustment

- What are missing ?

Pass, Seeman, Shelat, *Analysis of the Blockchain Protocol in Asynchronous Networks*. Eurocrypt 2017.
Garay, Kiayias, Leonardos, *The Bitcoin Backbone Protocol with Chains of Variable Difficulty*. Crypto 2017.

Multi-mode systems

[Duong, **Z**, Chepurnoy 17]

- Full mode
- light modes (SPV, prune,…)
- Bitcoin is a multi-mode system by design

Duong, Zhou, Chepurnoy, *Multi-Mode Cryptocurrency Systems*. Manuscript.

Multi-mode systems

[Duong, **Z**, Chepurnoy 17]

- Why multi-mode ?

- How to define the security?

Duong, Zhou, Chepurnoy, *Multi-Mode Cryptocurrency Systems*. Manuscript.

Alternative Mechanisms

A Unified View

We don't want to put all eggs in one basket.

Can we do a better job than Nakamoto?

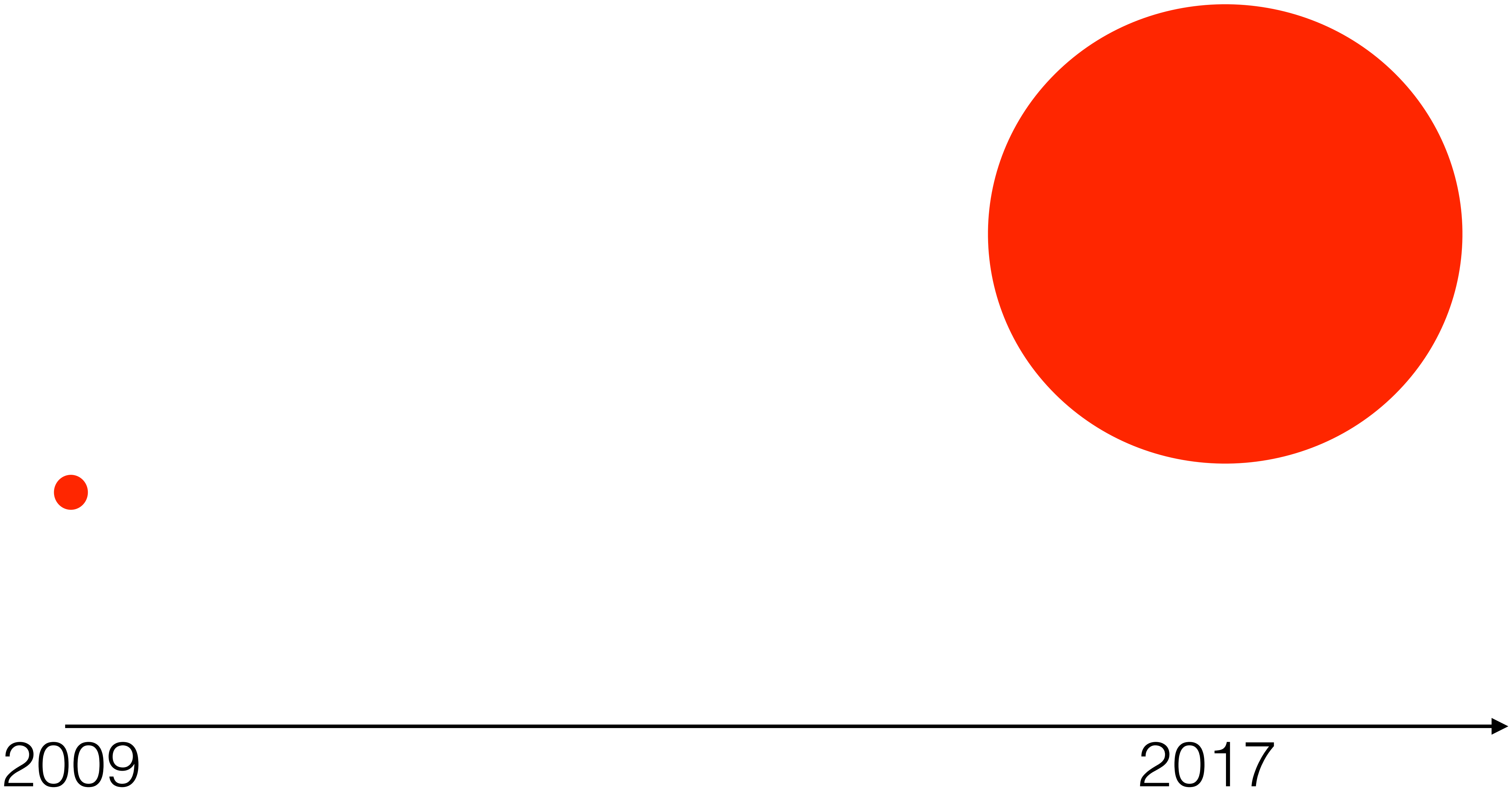
Nakamoto’s design

- **Most of tools were known**
  - **Public keys as identities**
  - **Time stamping**
  - **Hash chain**
  - **Incentives**
  - **Proof-of-work**

Nakamoto's design

- Amazing design
  - Put them together
- Cute points
  - Open (via PoW); easy to join/leave
  - Suitable incentives
  - Adaptive difficulty adjustment
  - Scalable to a huge network of nodes; very lightweight communication

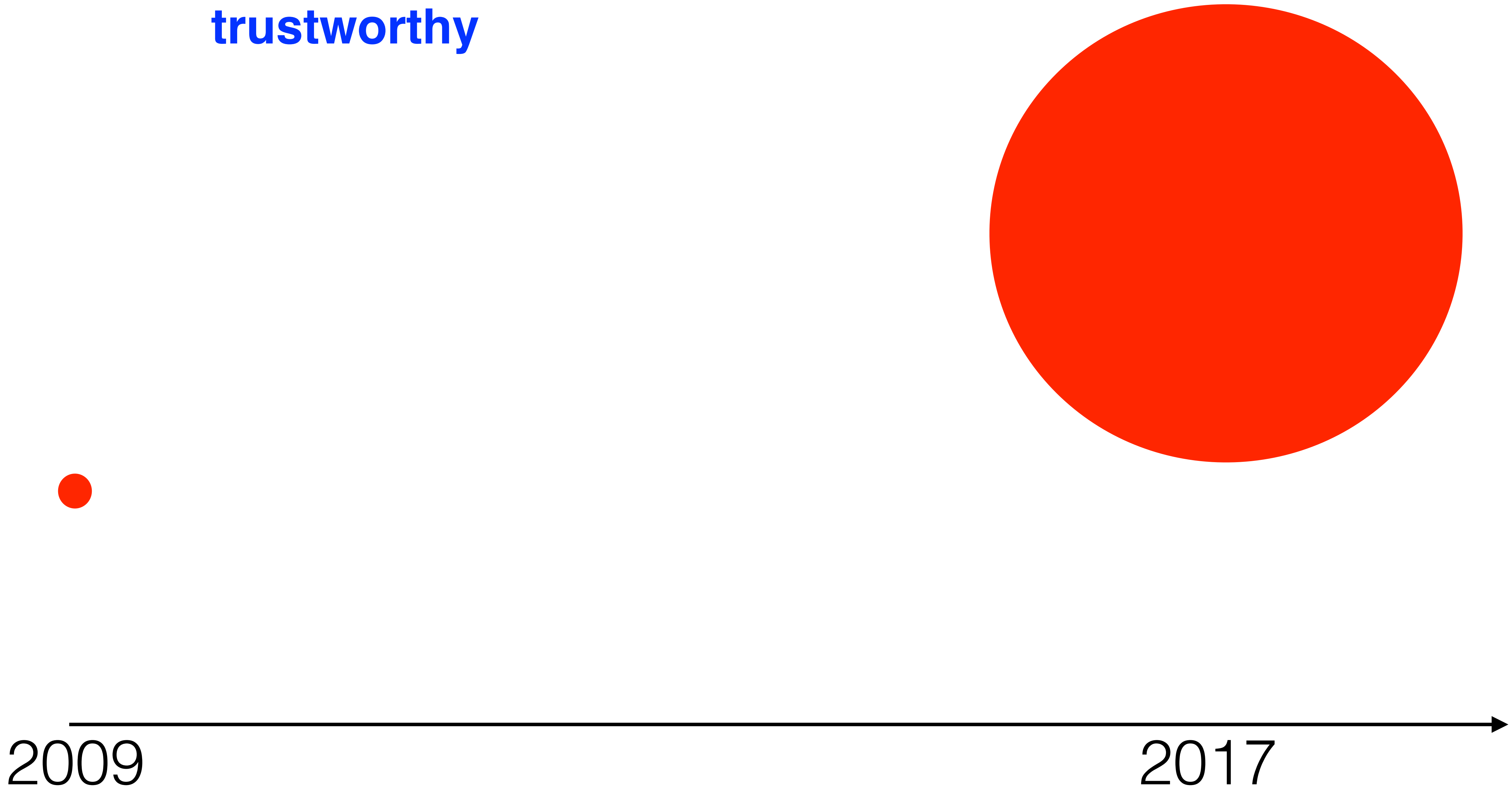
Nakamoto’s design



Nakamoto’s design

the blockchain is

backed up by a huge network of computing power; censorship resilient; very trustworthy



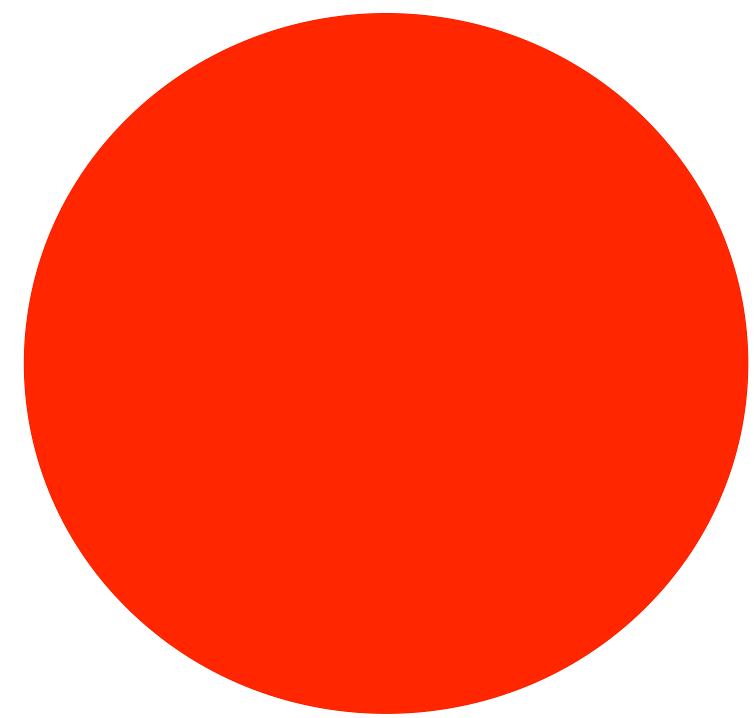
Nakamoto’s design

the blockchain is

backed up by a huge network of computing power; censorship resilient; very trustworthy

The flip side:

lots of electricity has been invested in this system; not environment friendly



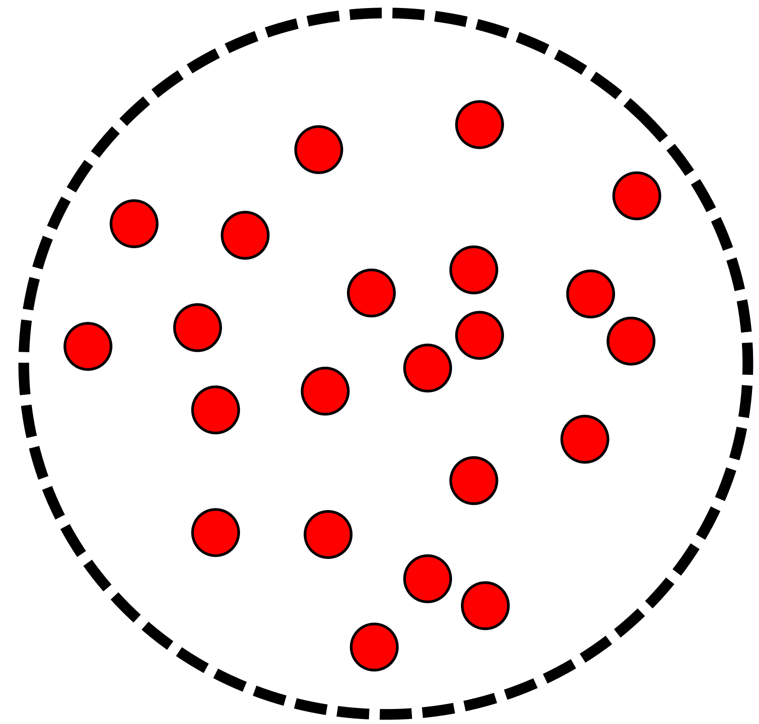
Nakamoto's design

- Cute points
  - Open (via PoW)
  - Suitable incentives
  - Adaptive difficulty adjustment
  - Scalable to a huge network of nodes; very lightweight communication

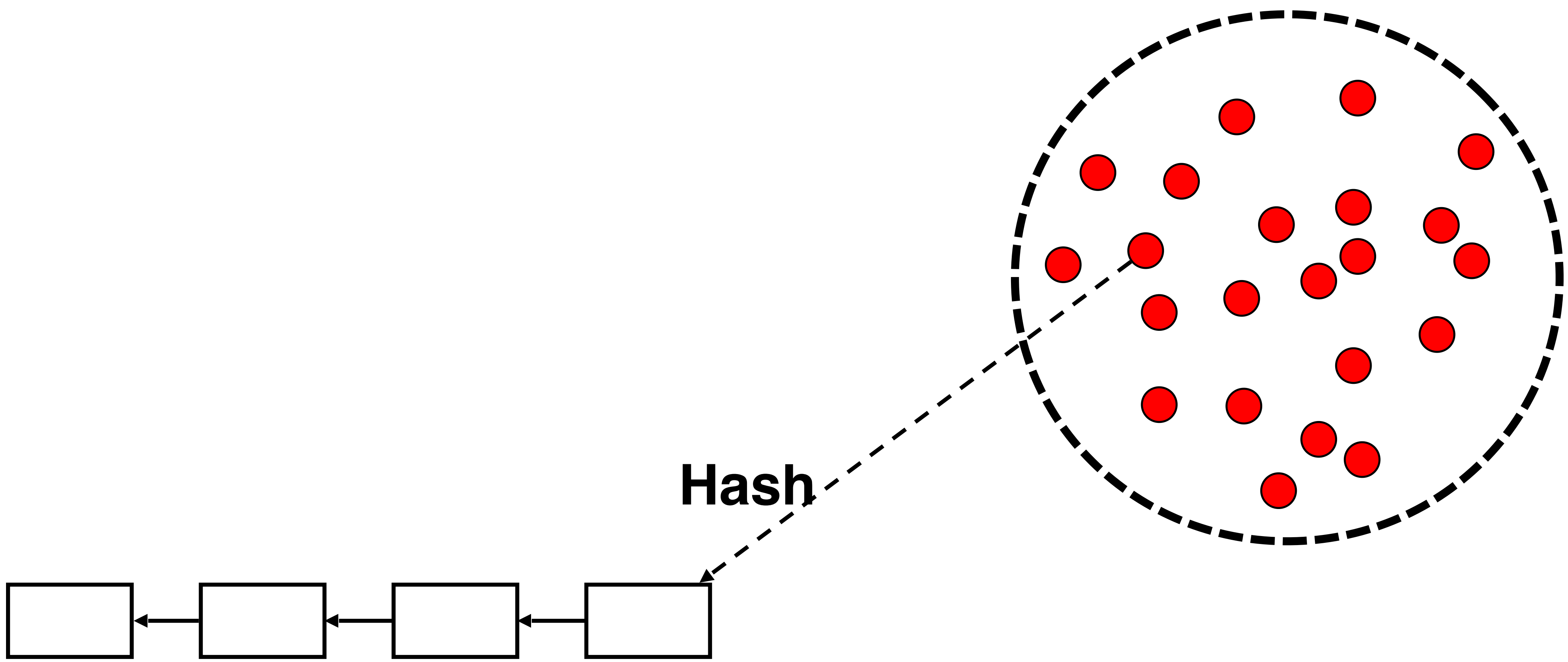
Nakamoto's design

- Cute points
  - Open (via PoW)
    tricky; overlooked
  - Suitable incentives
  - Adaptive difficulty adjustment
    [GKL17]
  - Scalable to a huge network of nodes; very lightweight communication
    tricky; overlooked

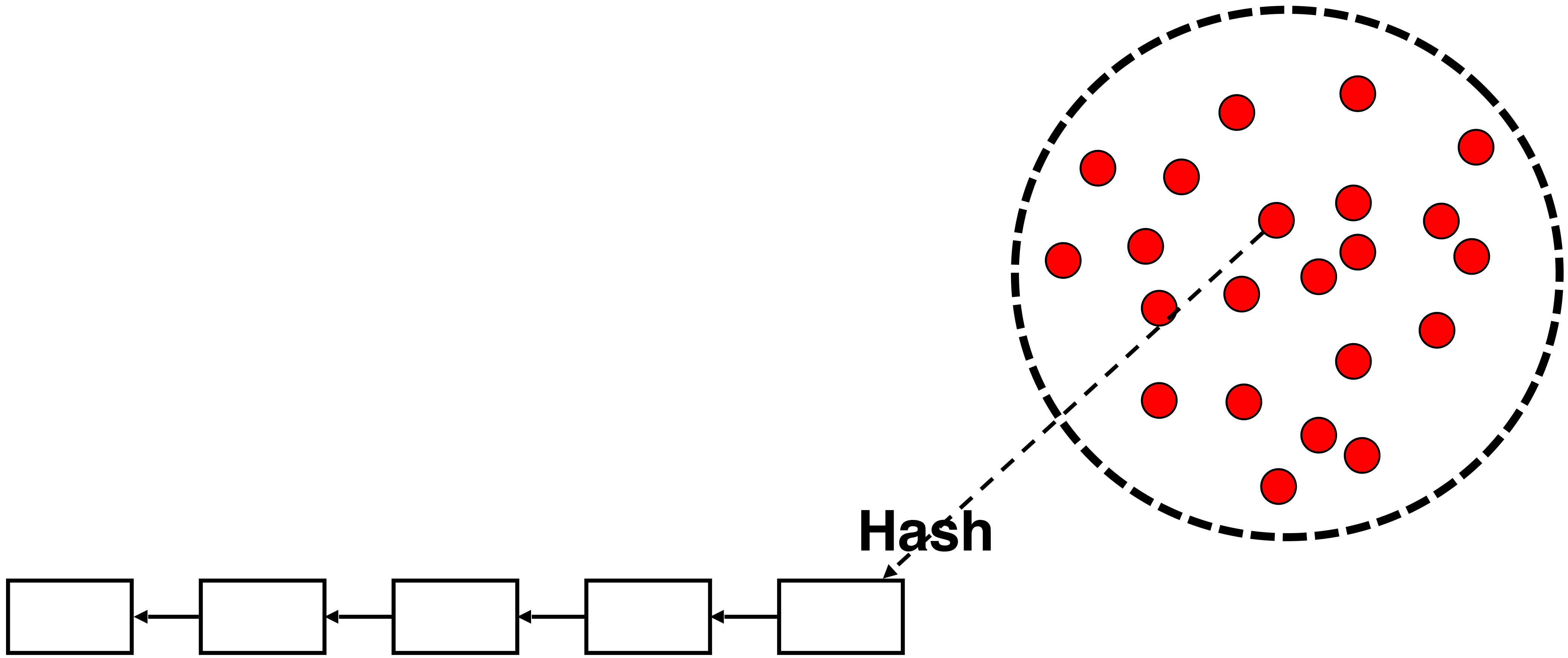
Nakamoto’s design



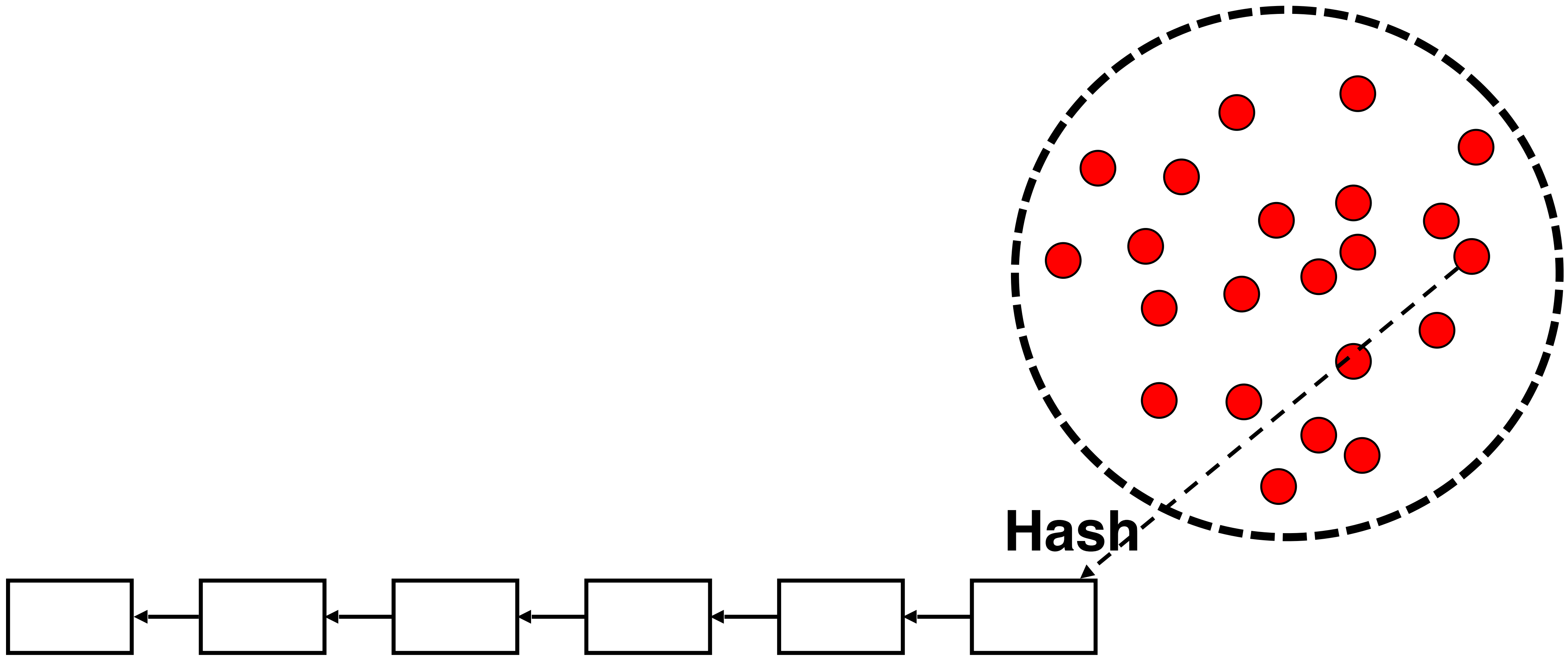
Nakamoto's design



Nakamoto’s design

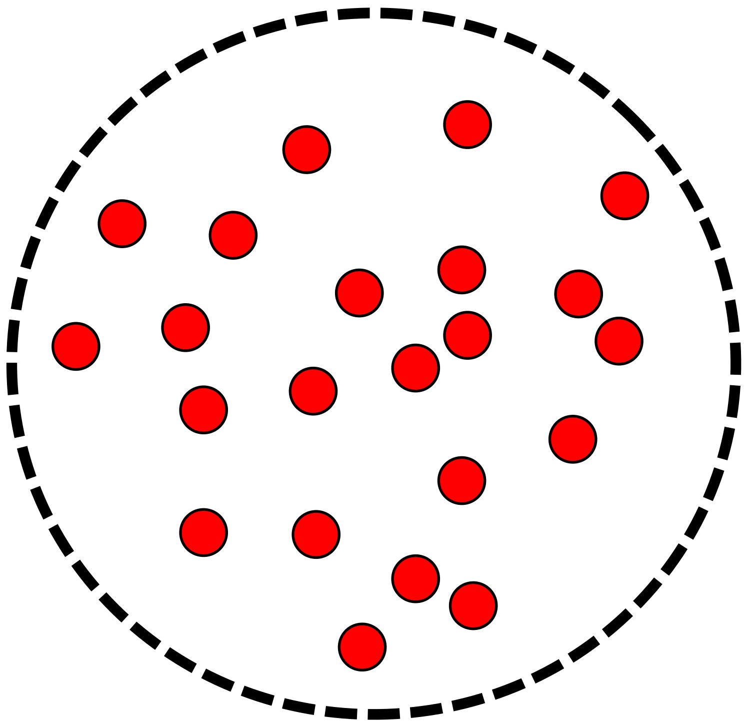


Nakamoto's design



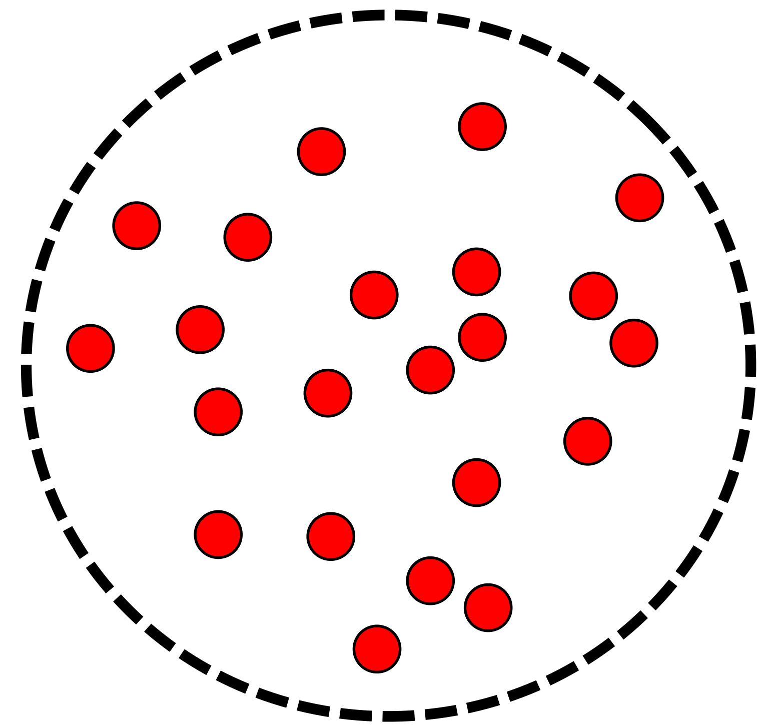
Nakamoto's design

Alternative View



Alternative View

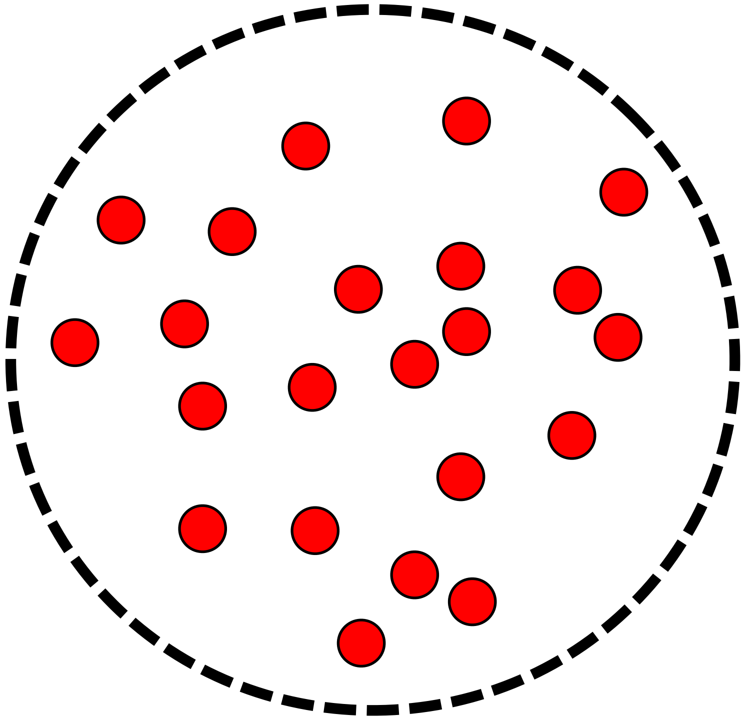
Nakamoto's design



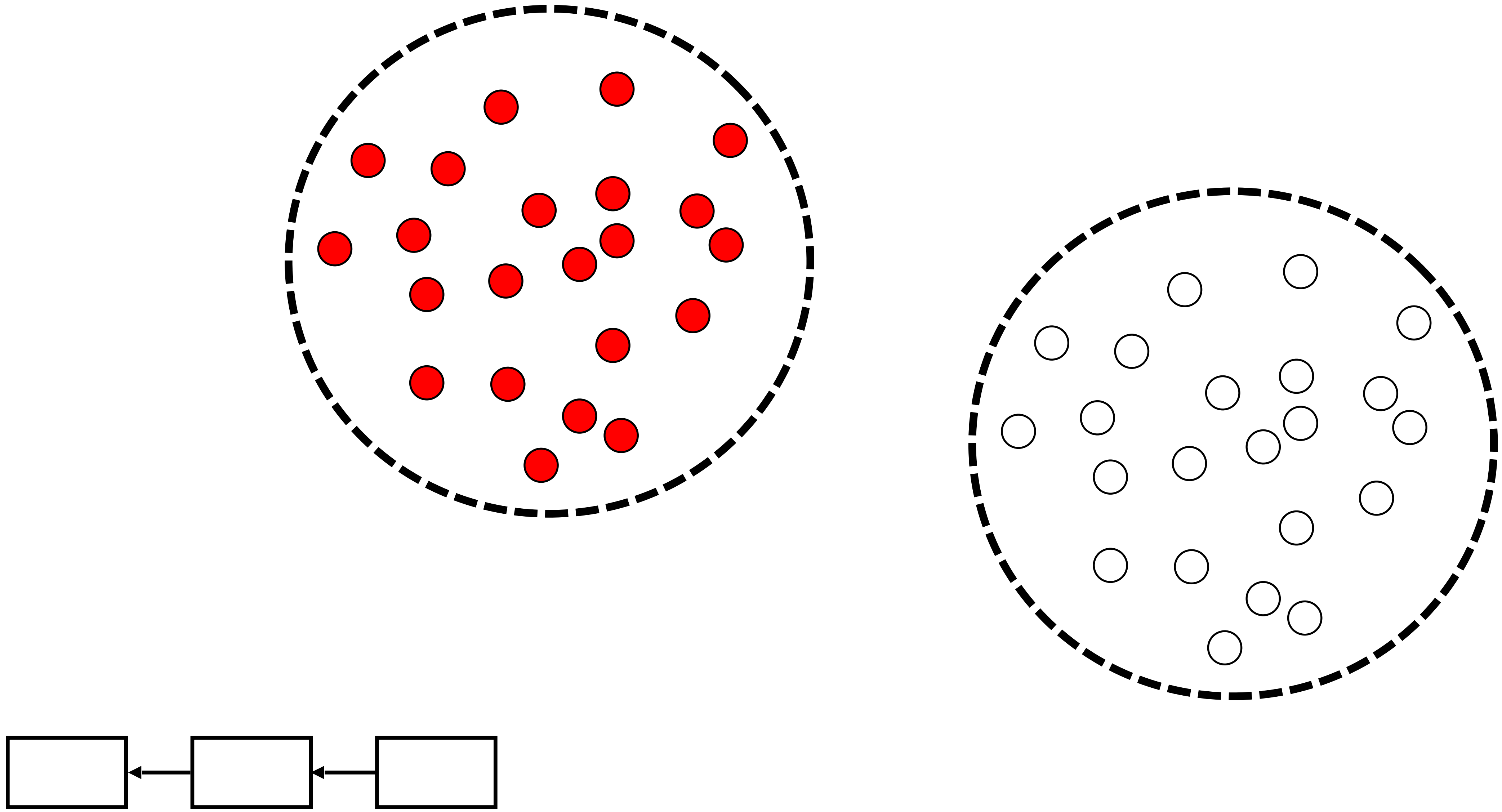
(inspired by ideas in [Garay, Kiayias, **Z**., CSF10])

Garay, Kiayias, Zhou, *A Framework for the Sound Specification of Cryptographic Tasks*. CSF 2010

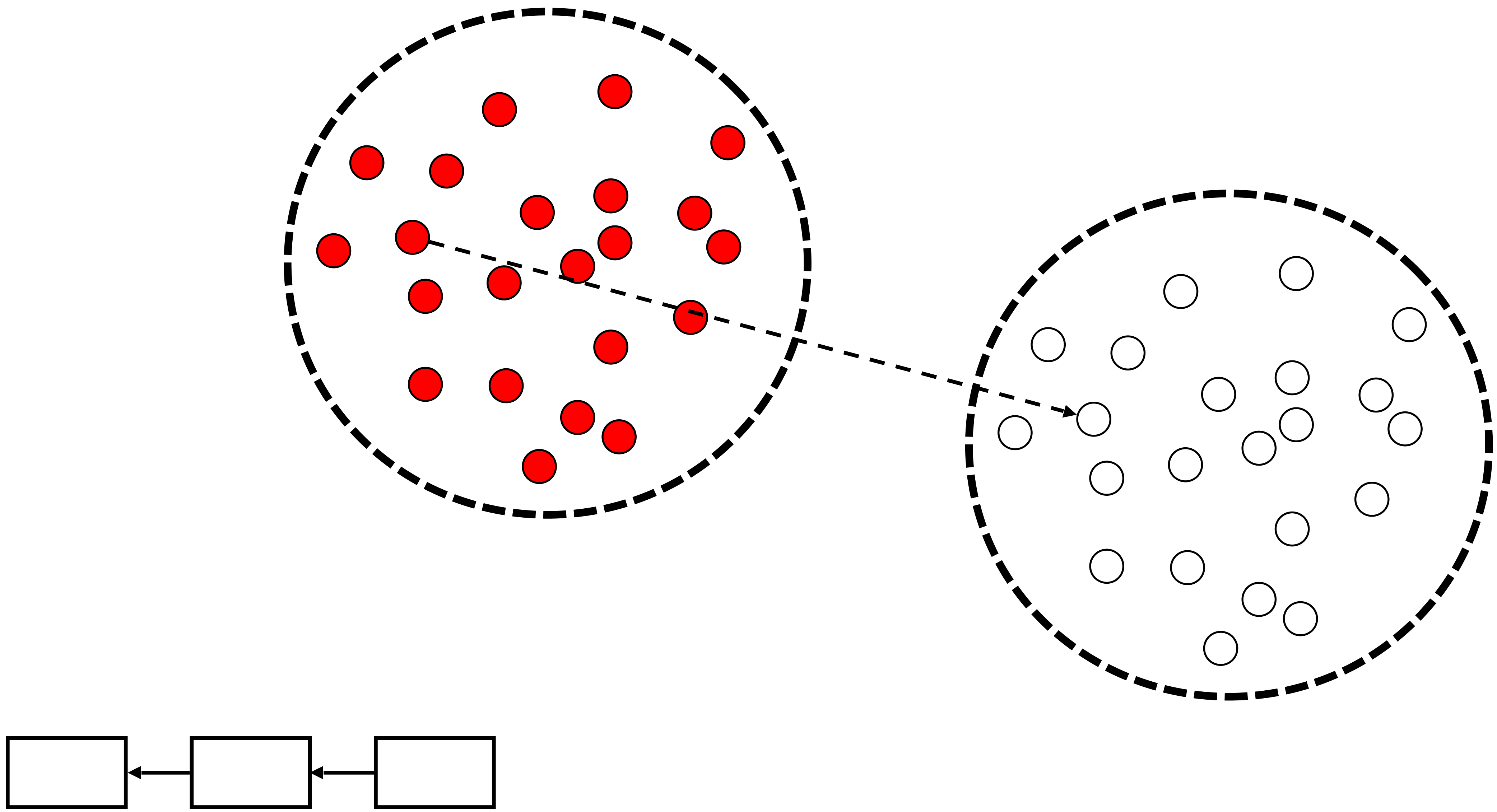
Nakamoto Blockchain: Alternative View



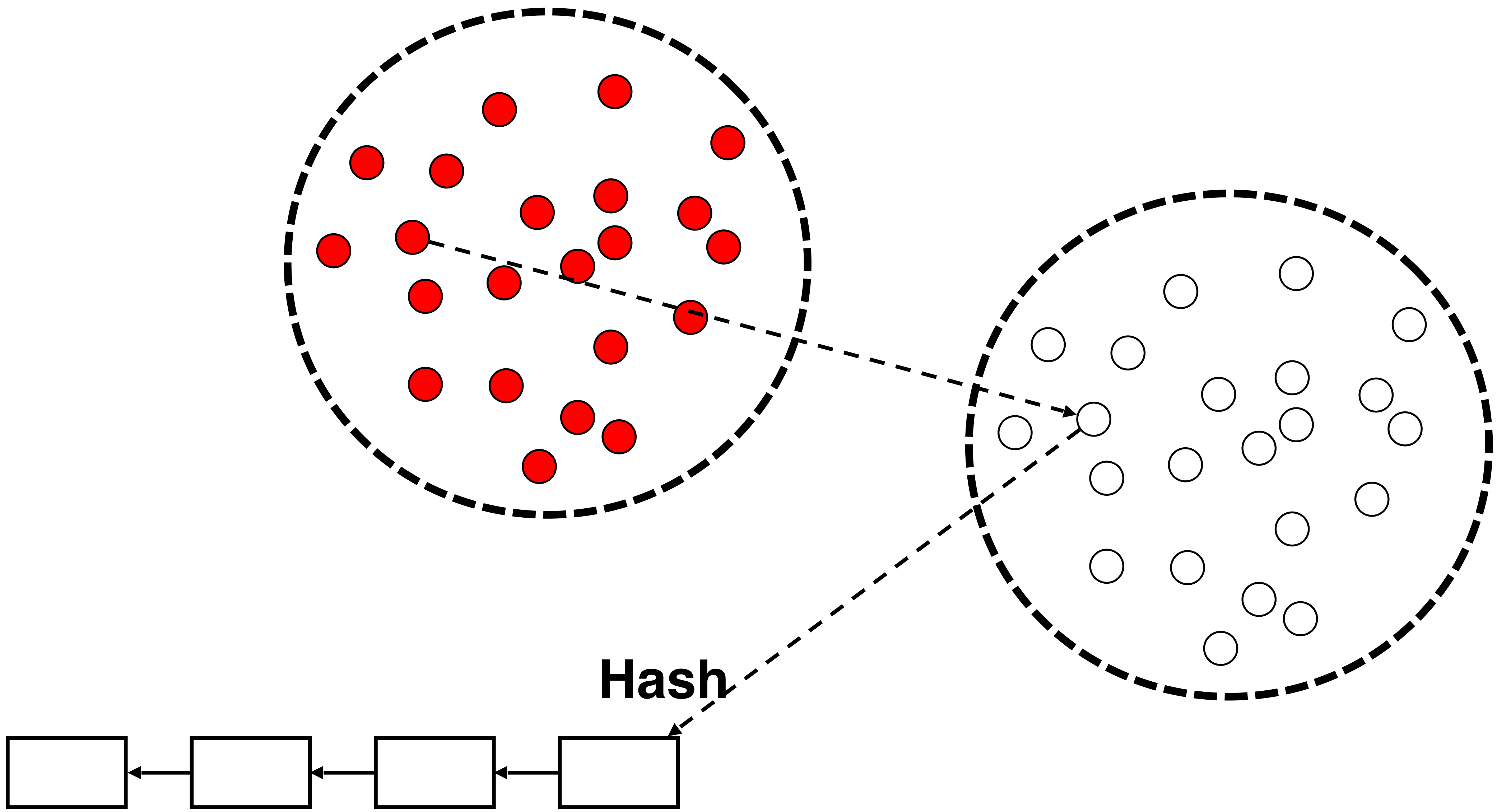
Nakamoto Blockchain: Alternative View



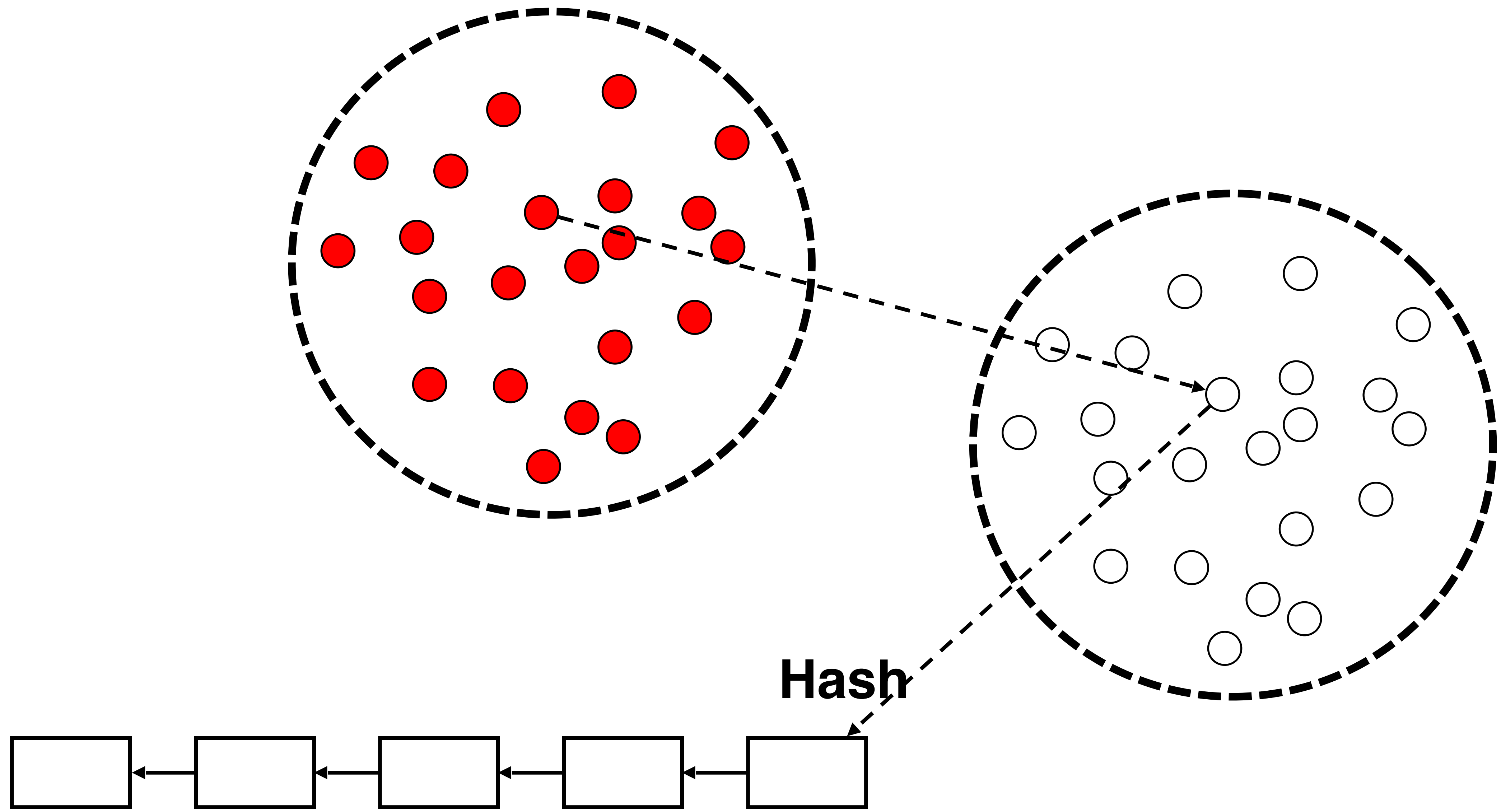
Nakamoto Blockchain: Alternative View



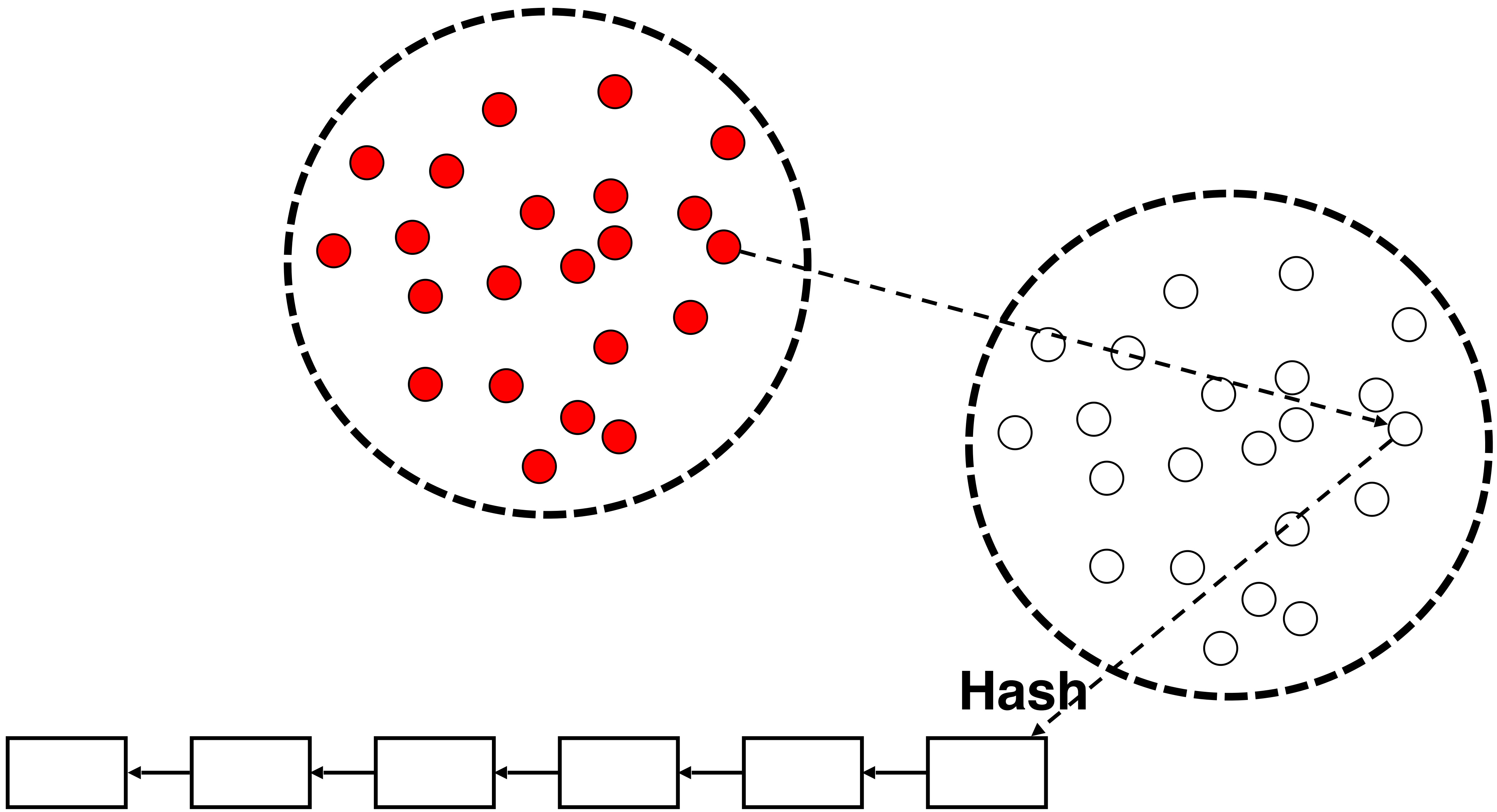
Nakamoto Blockchain: Alternative View



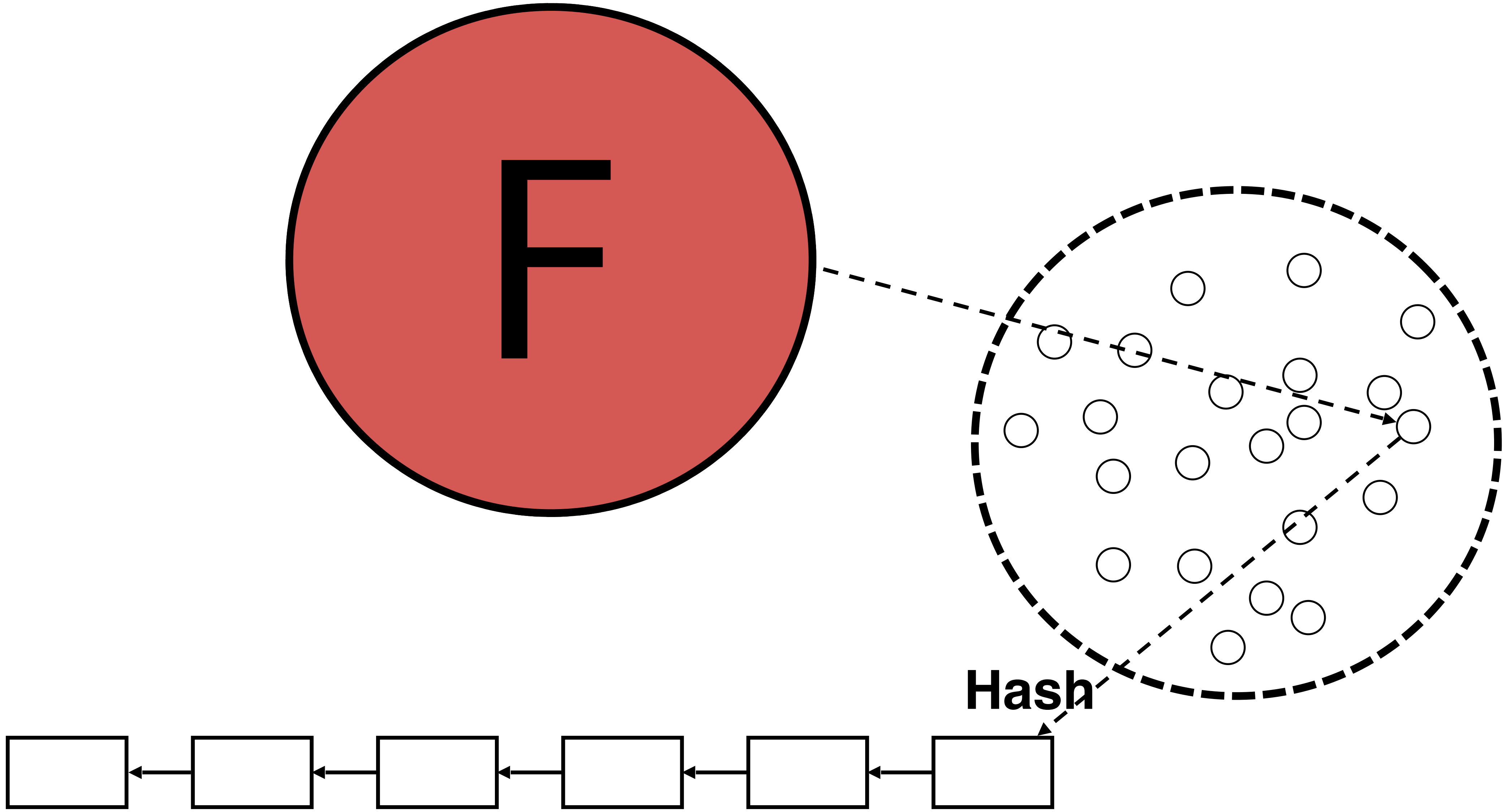
Nakamoto Blockchain: Alternative View

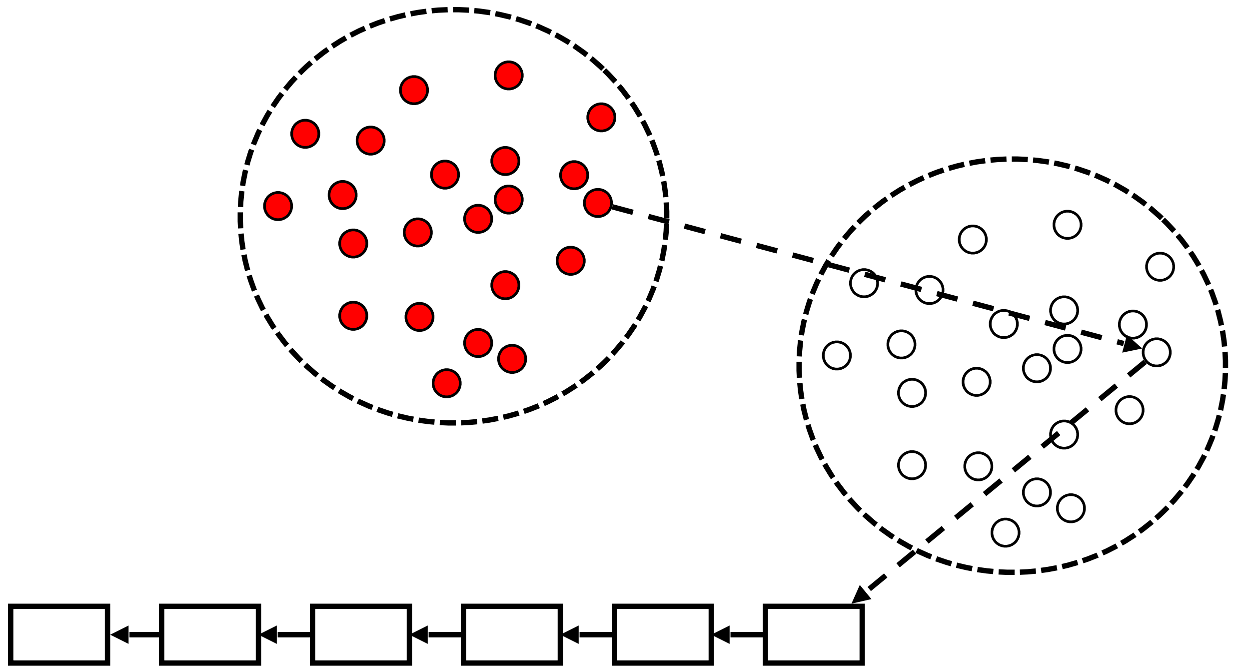


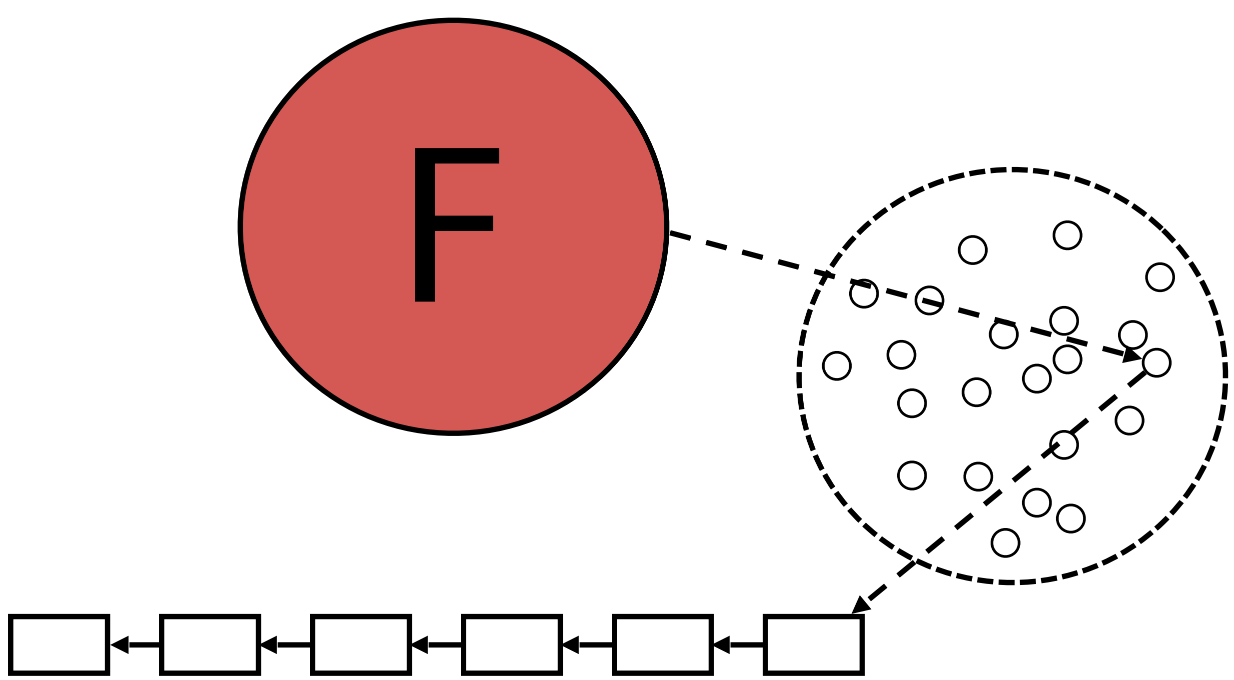
Nakamoto Blockchain: Alternative View



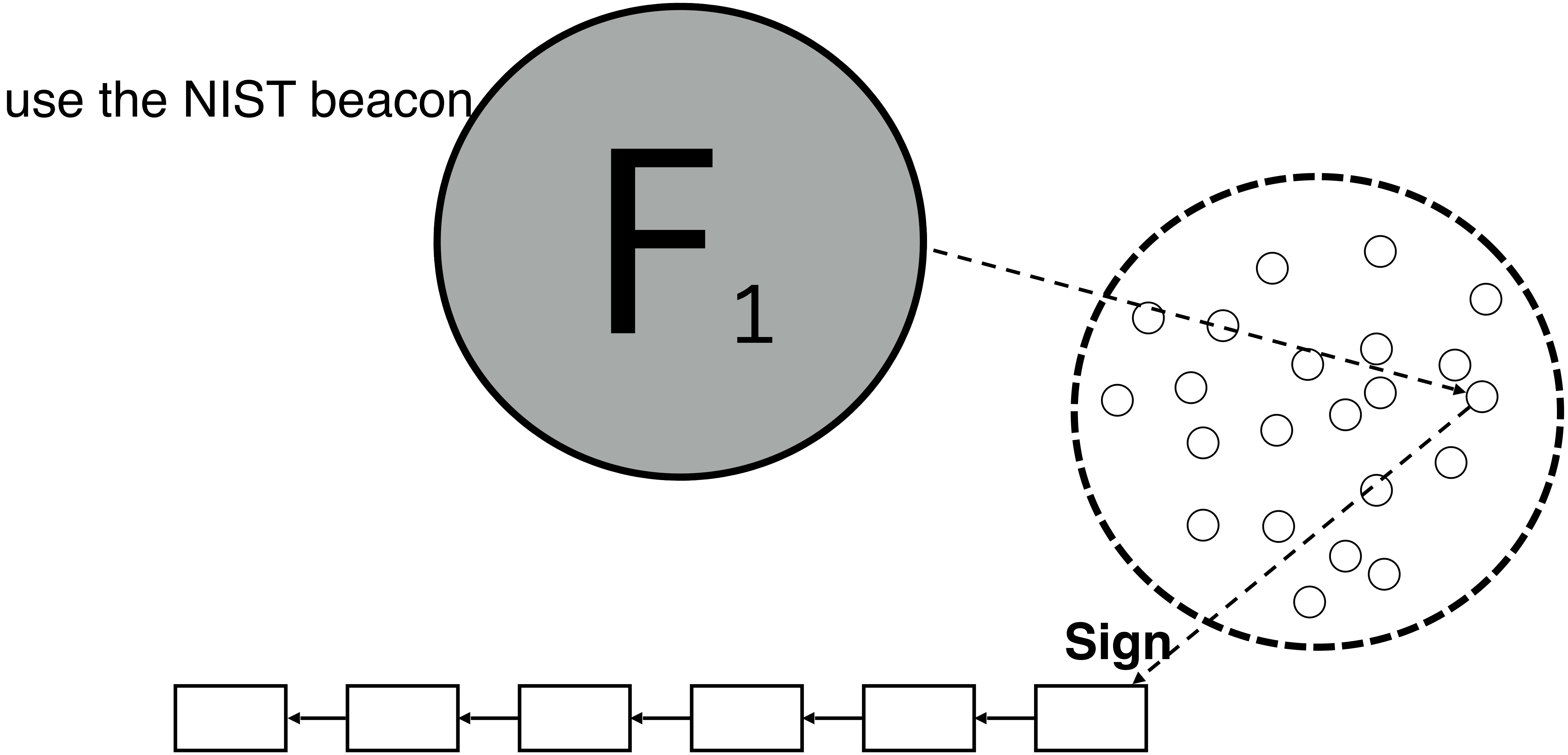
Nakamoto Blockchain: Alternative View





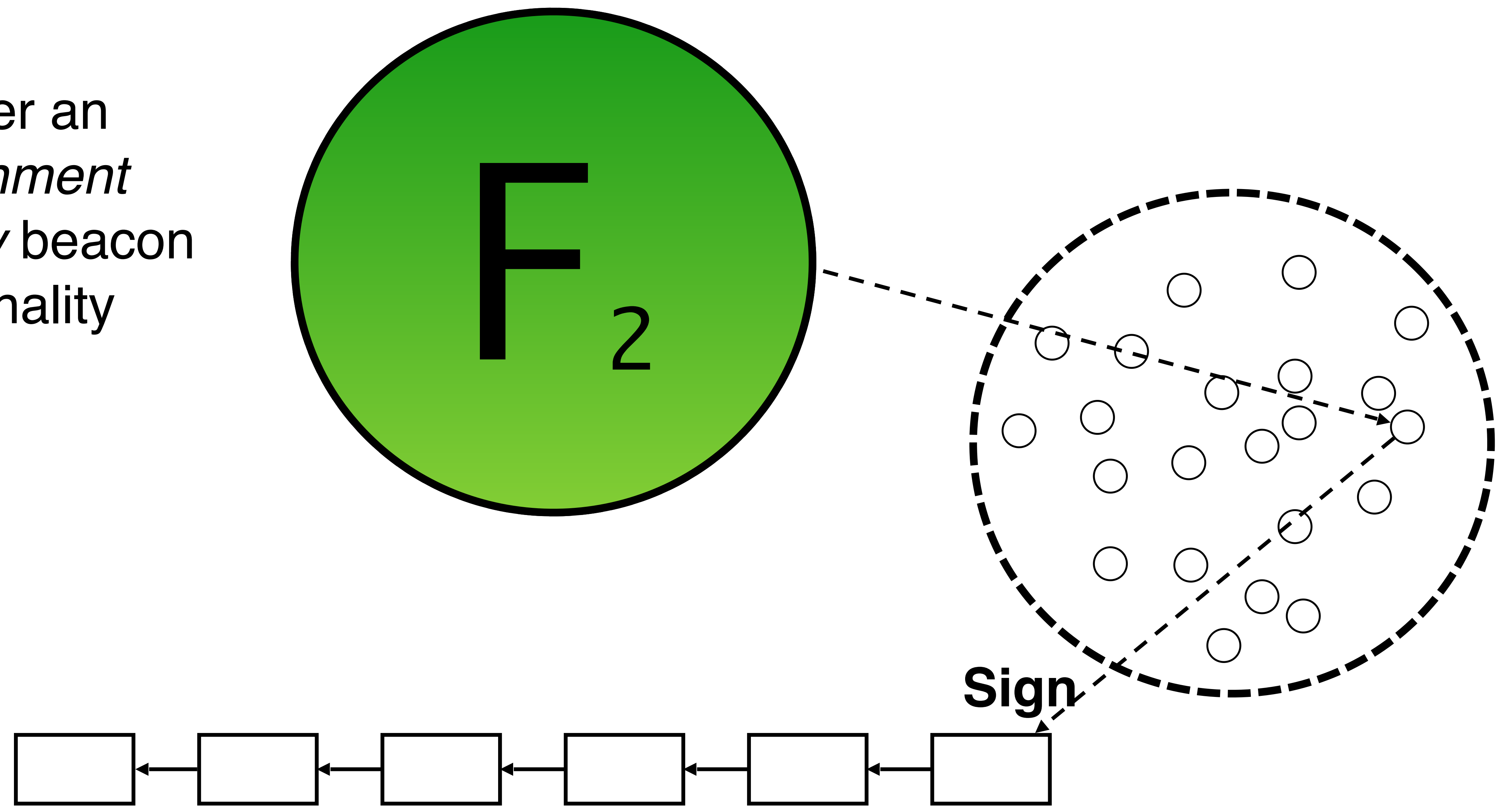

F

Beacon-based Blockchain



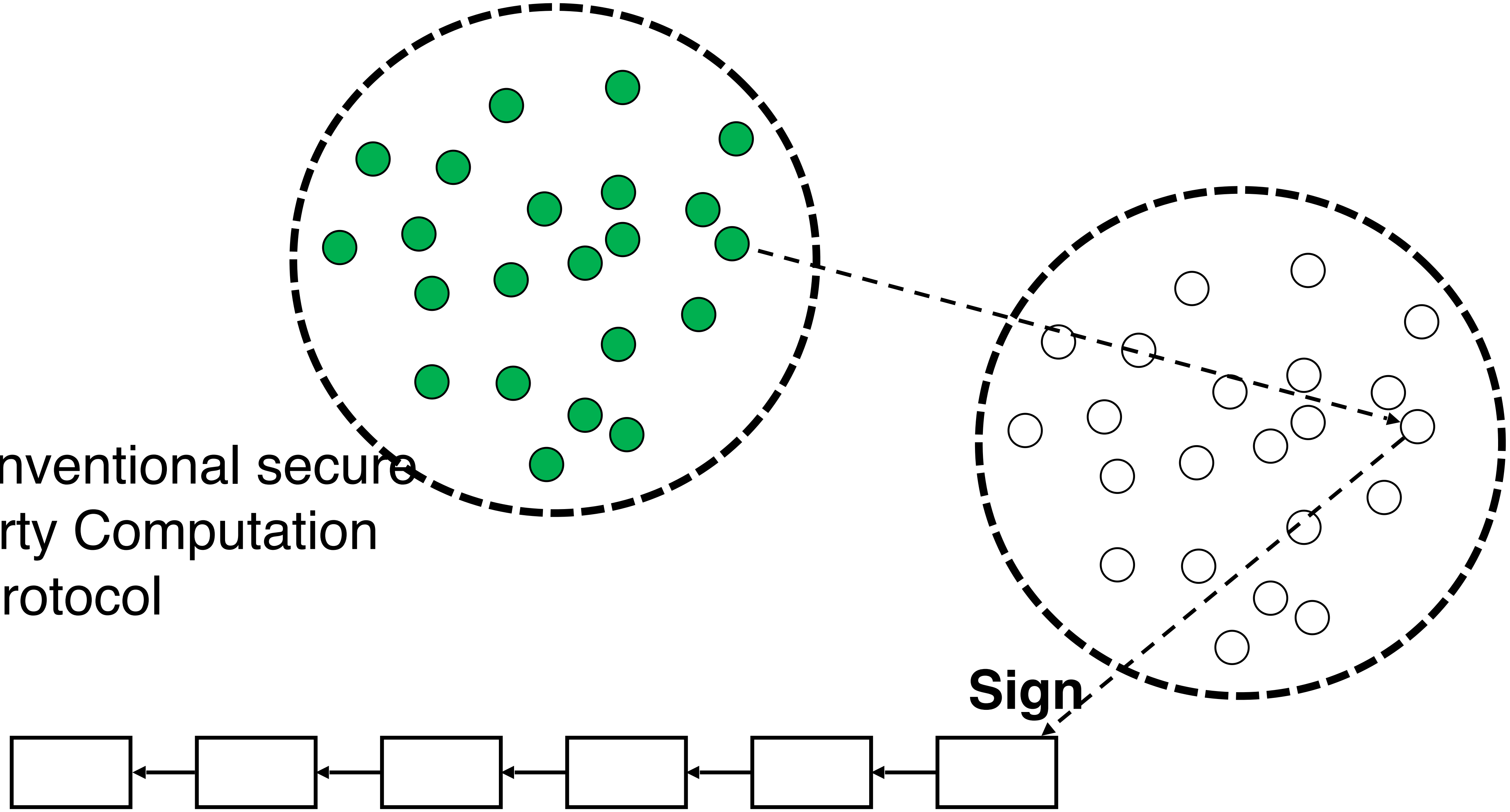
Beacon-based Blockchain

consider an *environment friendly* beacon functionality



Conventional MPC-based Blockchain

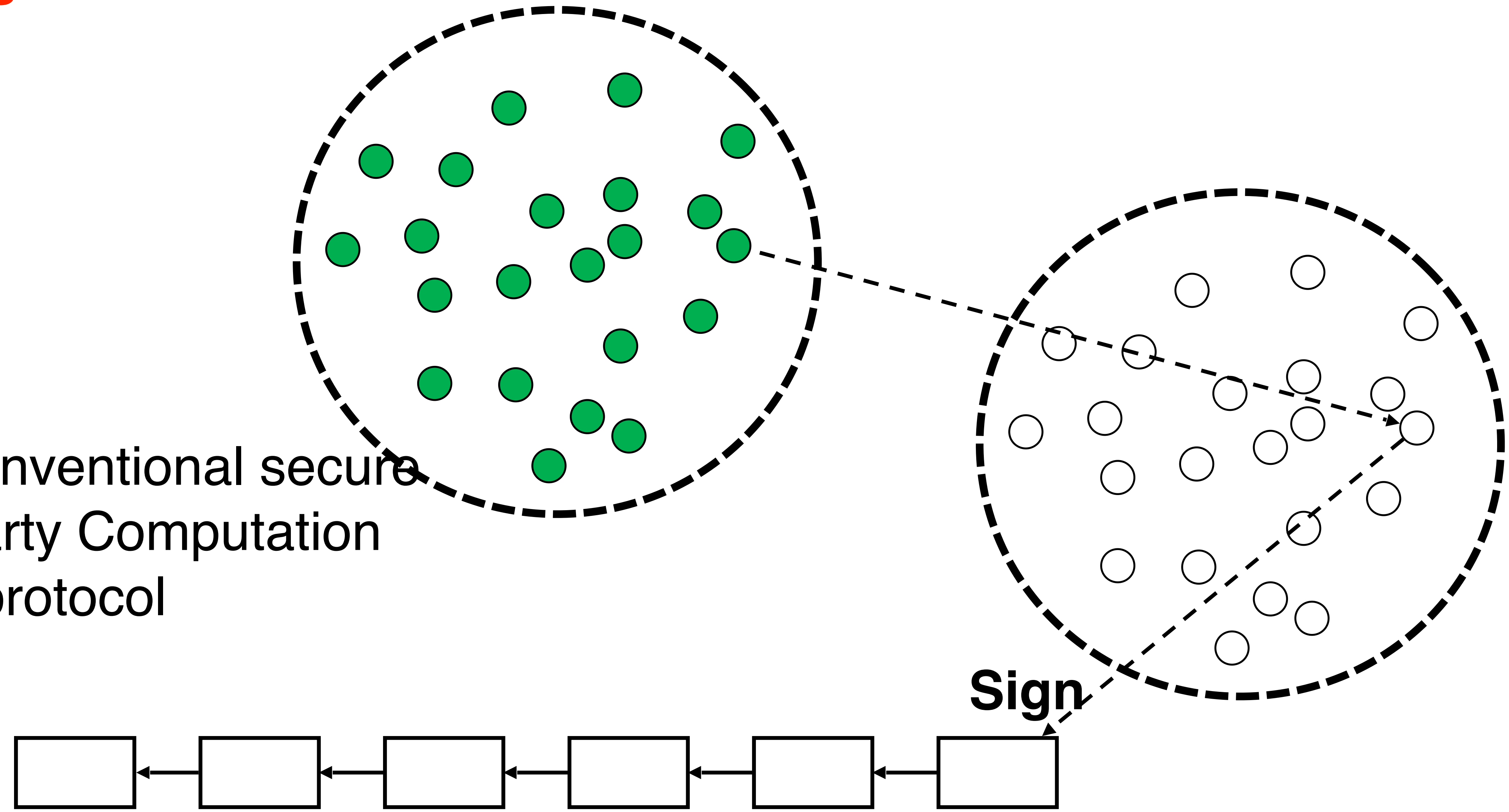
run a conventional secure Multi-Party Computation (MPC) protocol

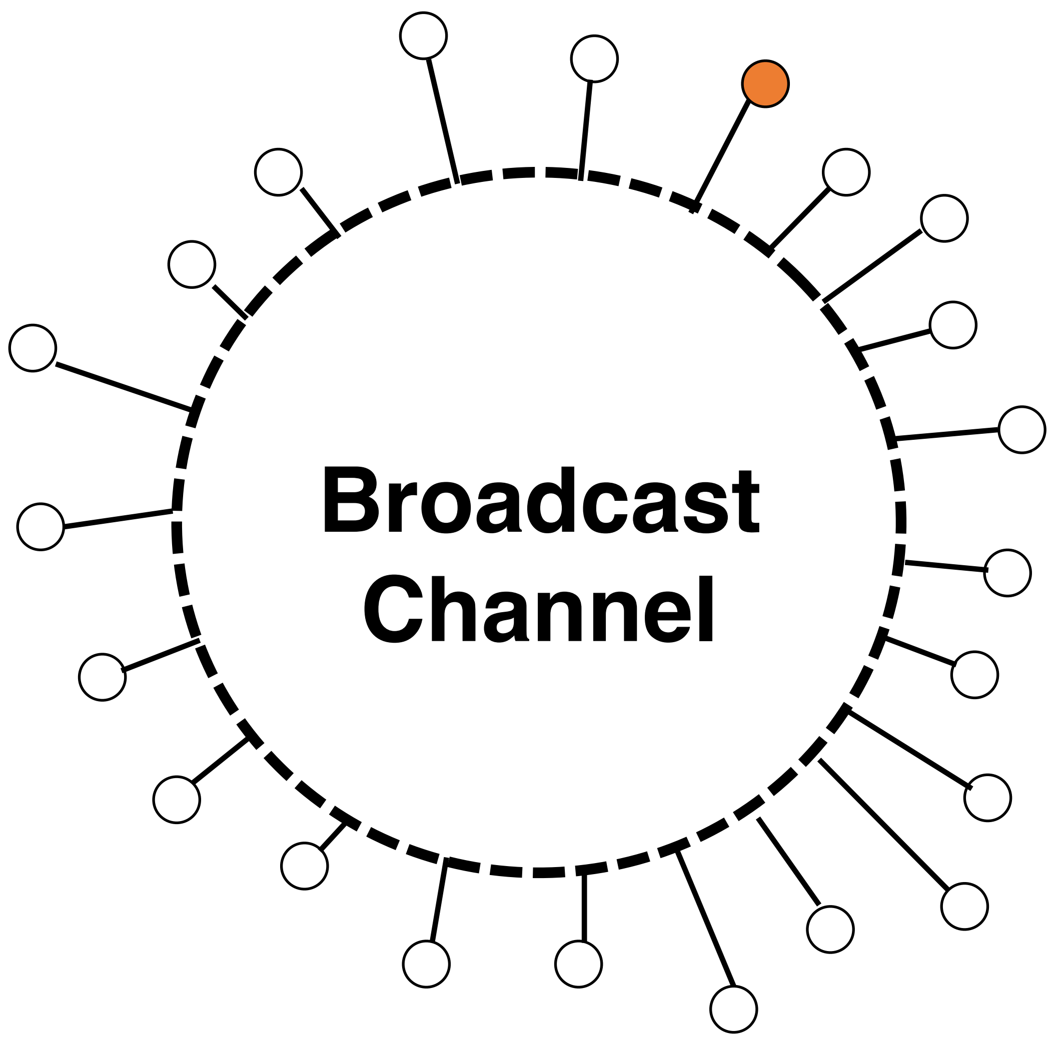


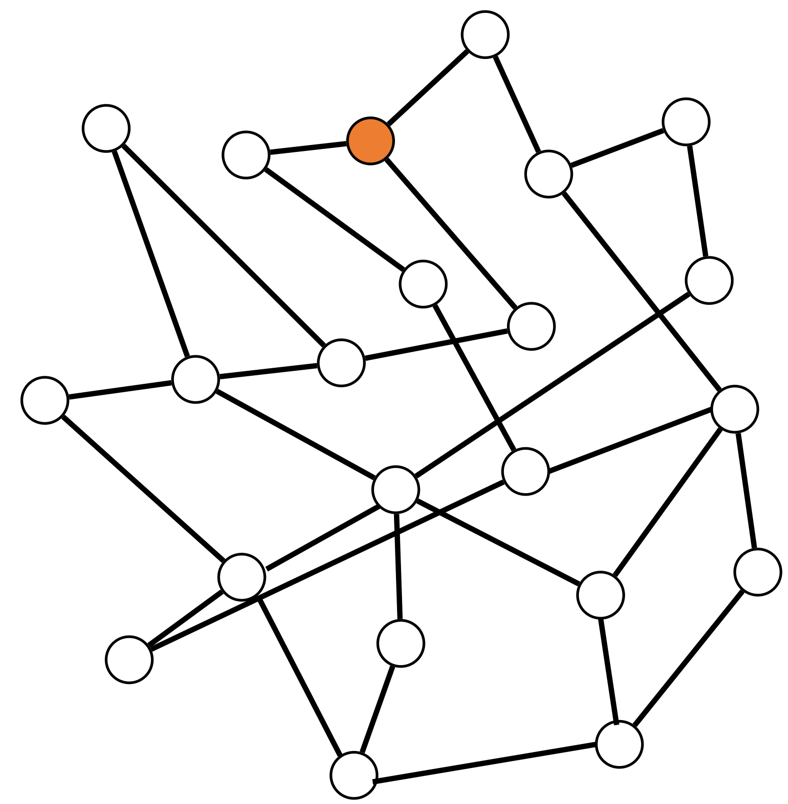
Conventional MPC-based Blockchain

our goal is to obtain a large-scale blockchain.
Warning: Conventional MPC cannot scale.

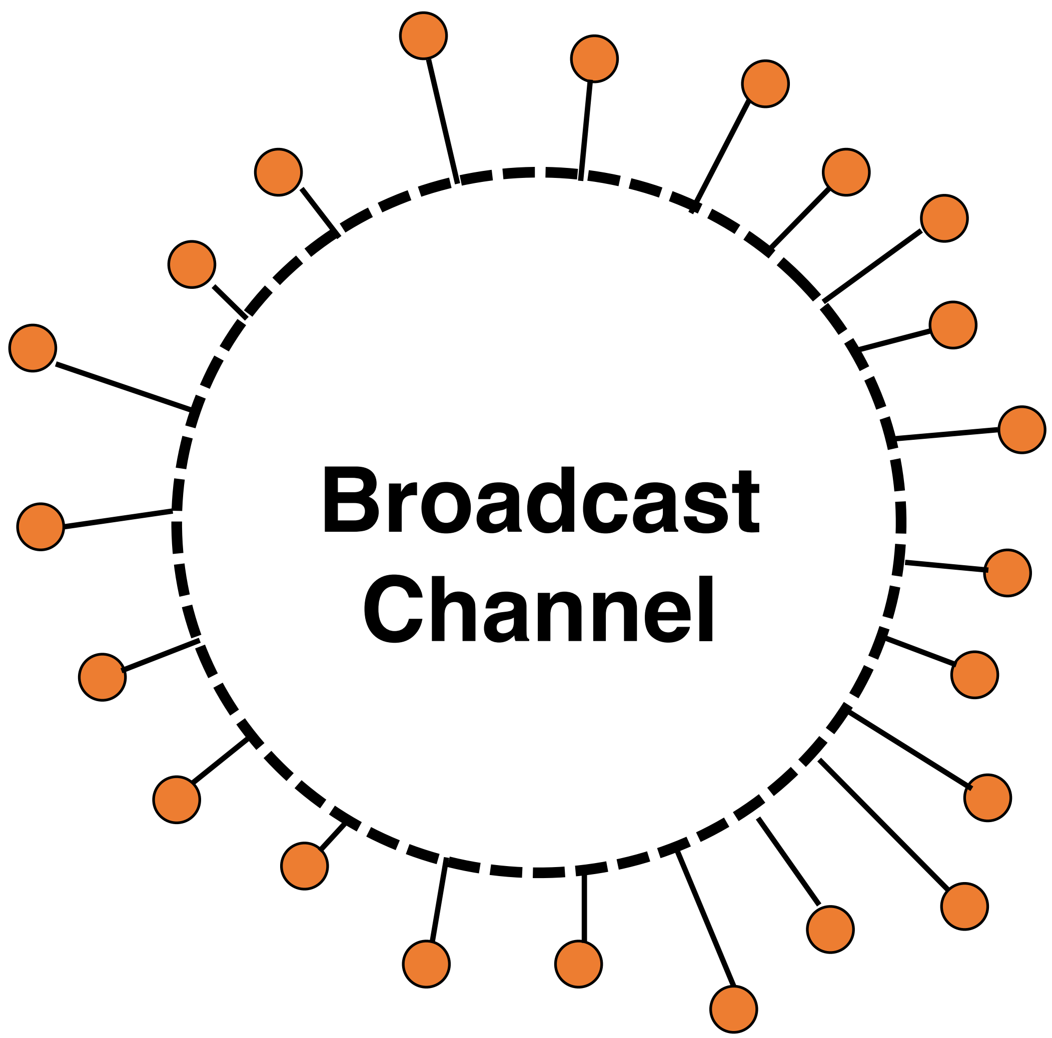
run a conventional secure Multi-Party Computation (MPC) protocol

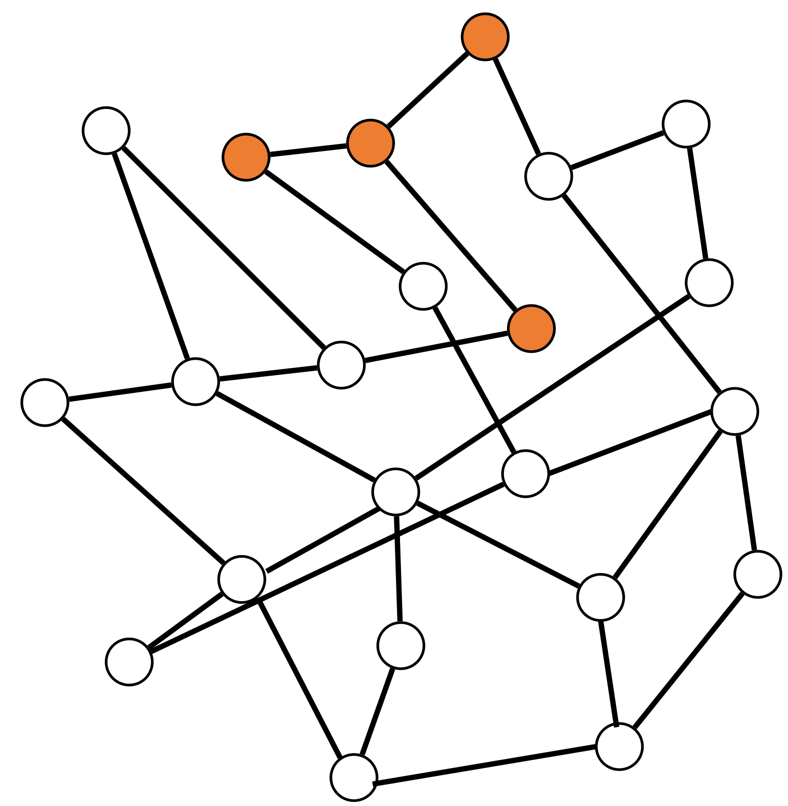


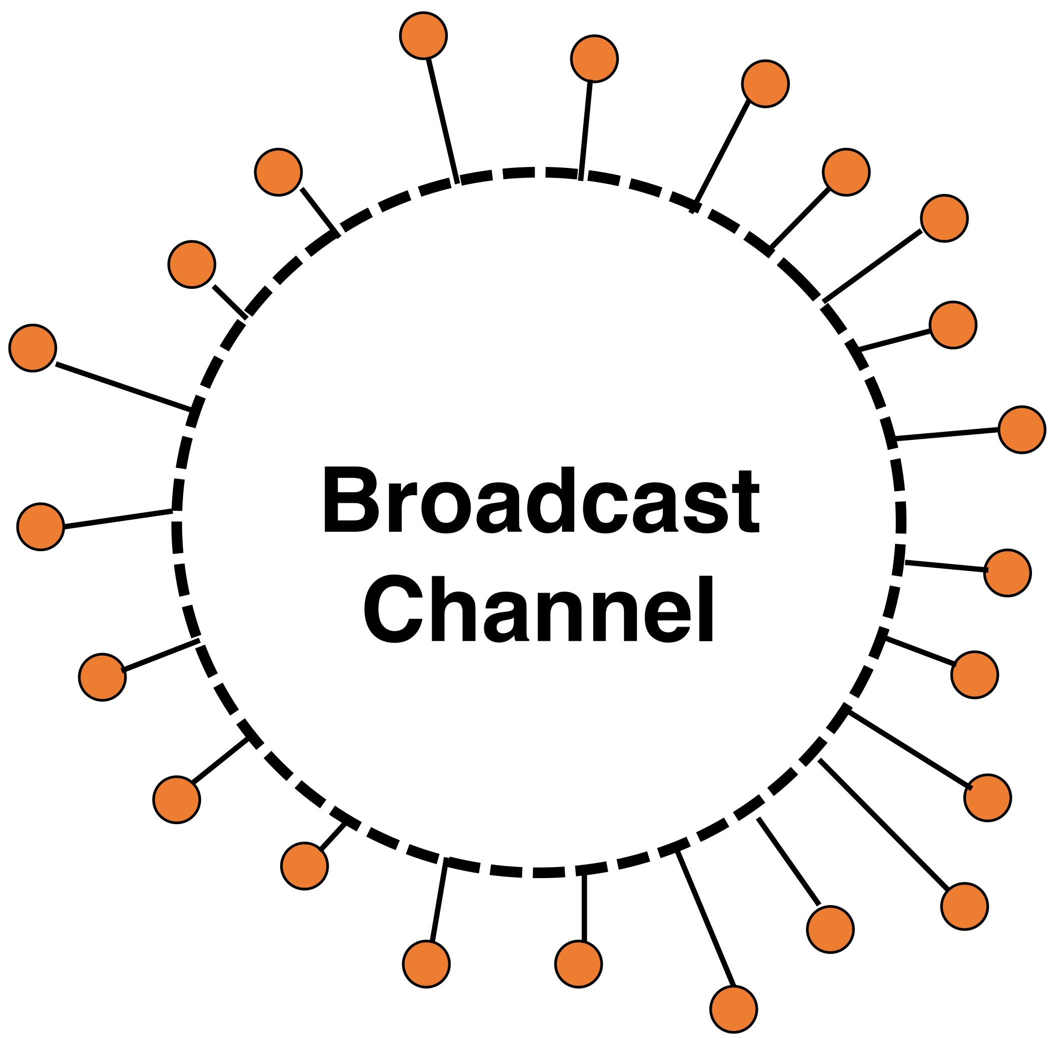


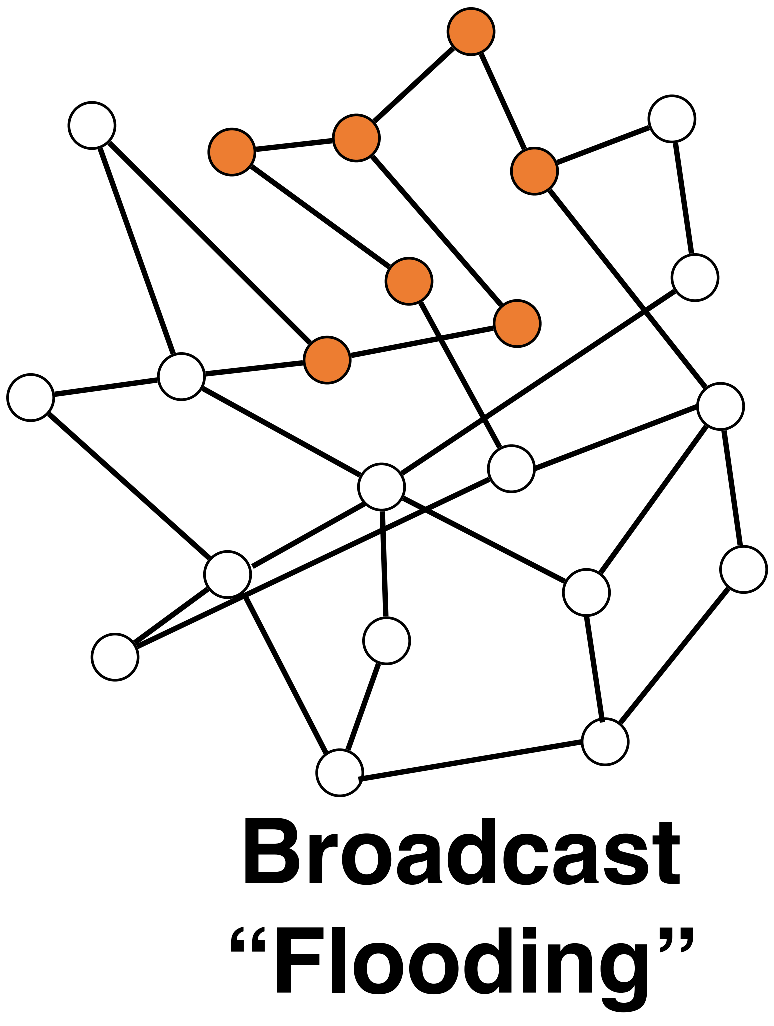


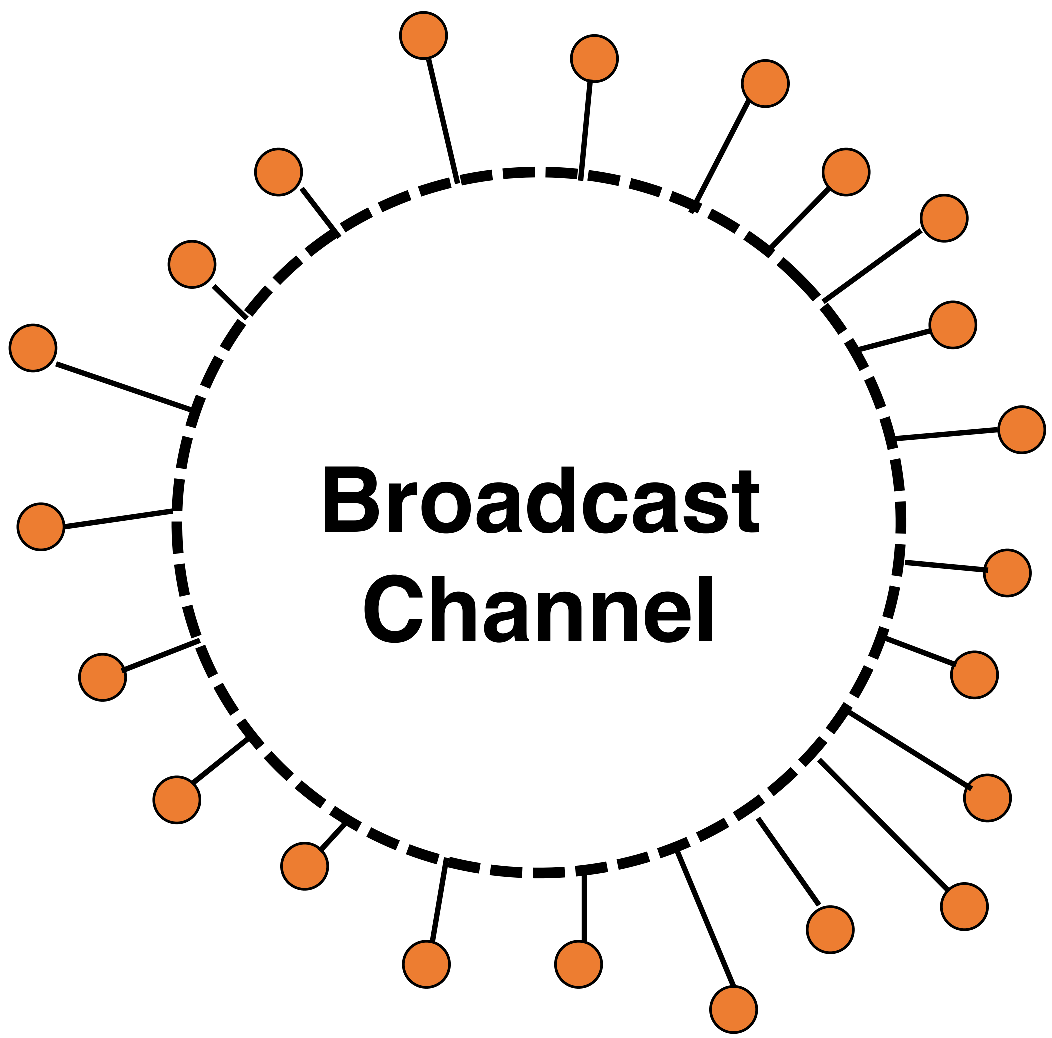
Broadcast
“Flooding”

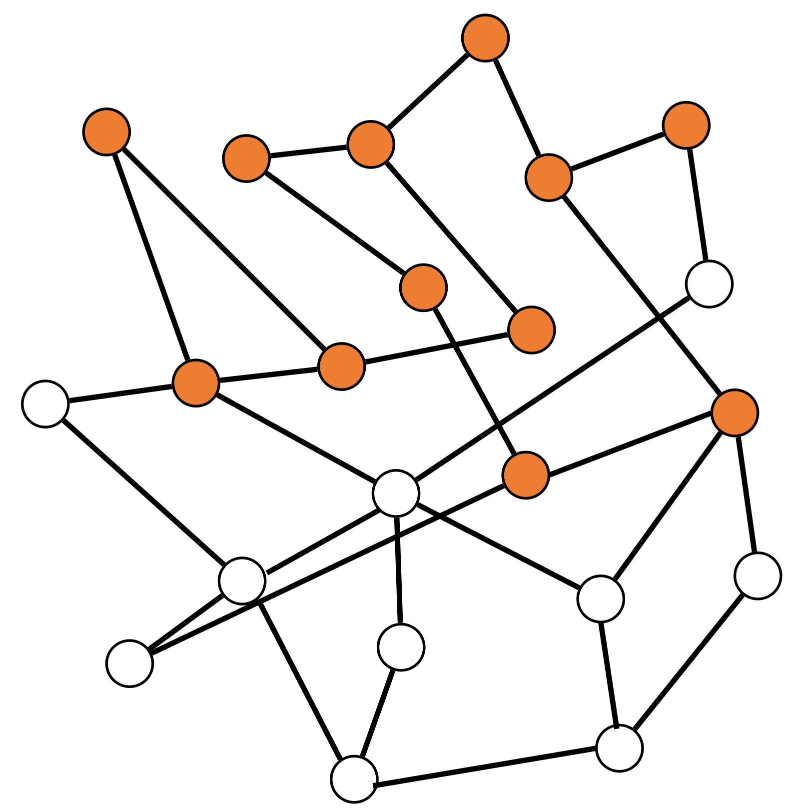

Broadcast
Channel


Broadcast
“Flooding”

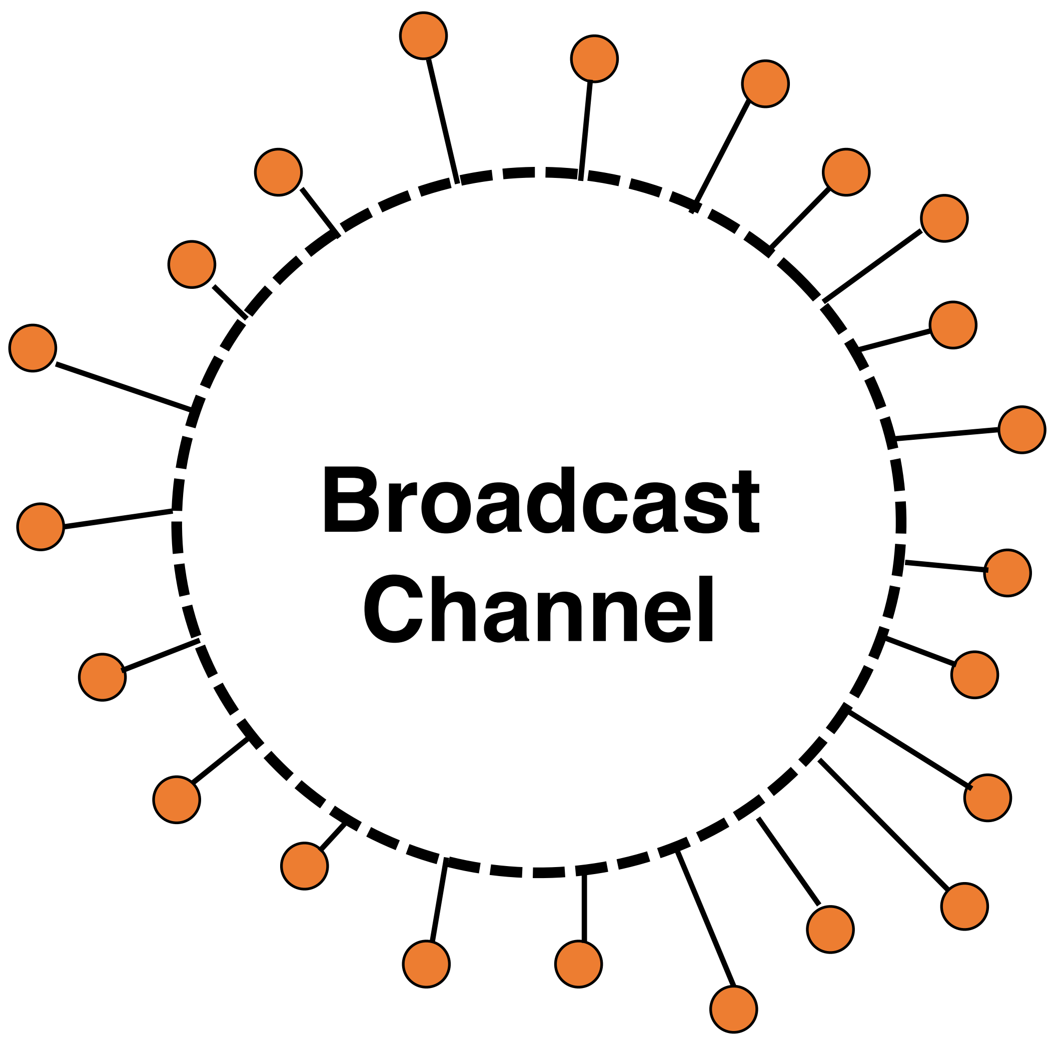

Broadcast
Channel

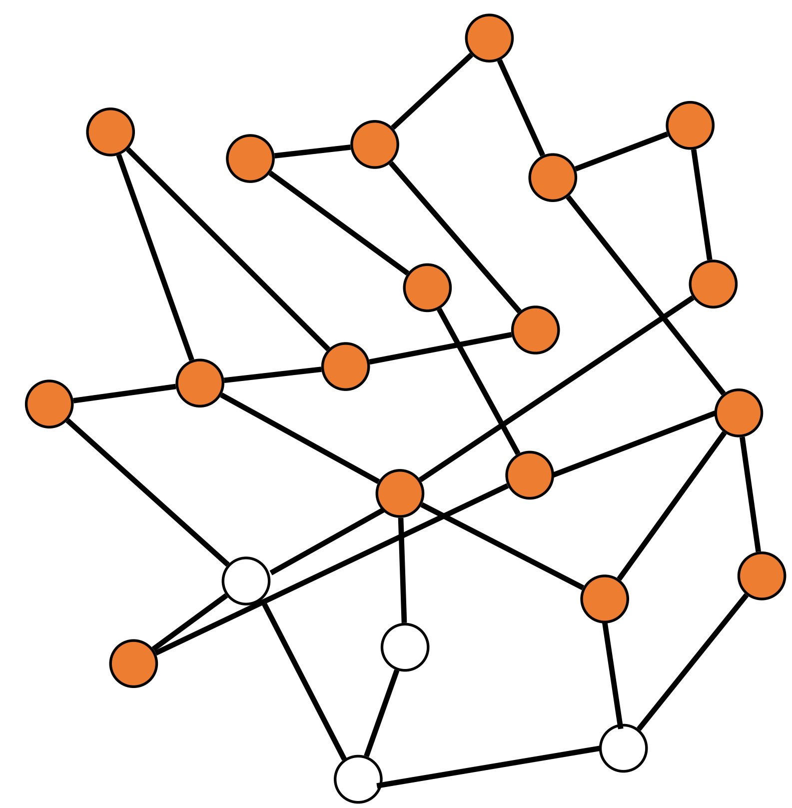

Broadcast
“Flooding”



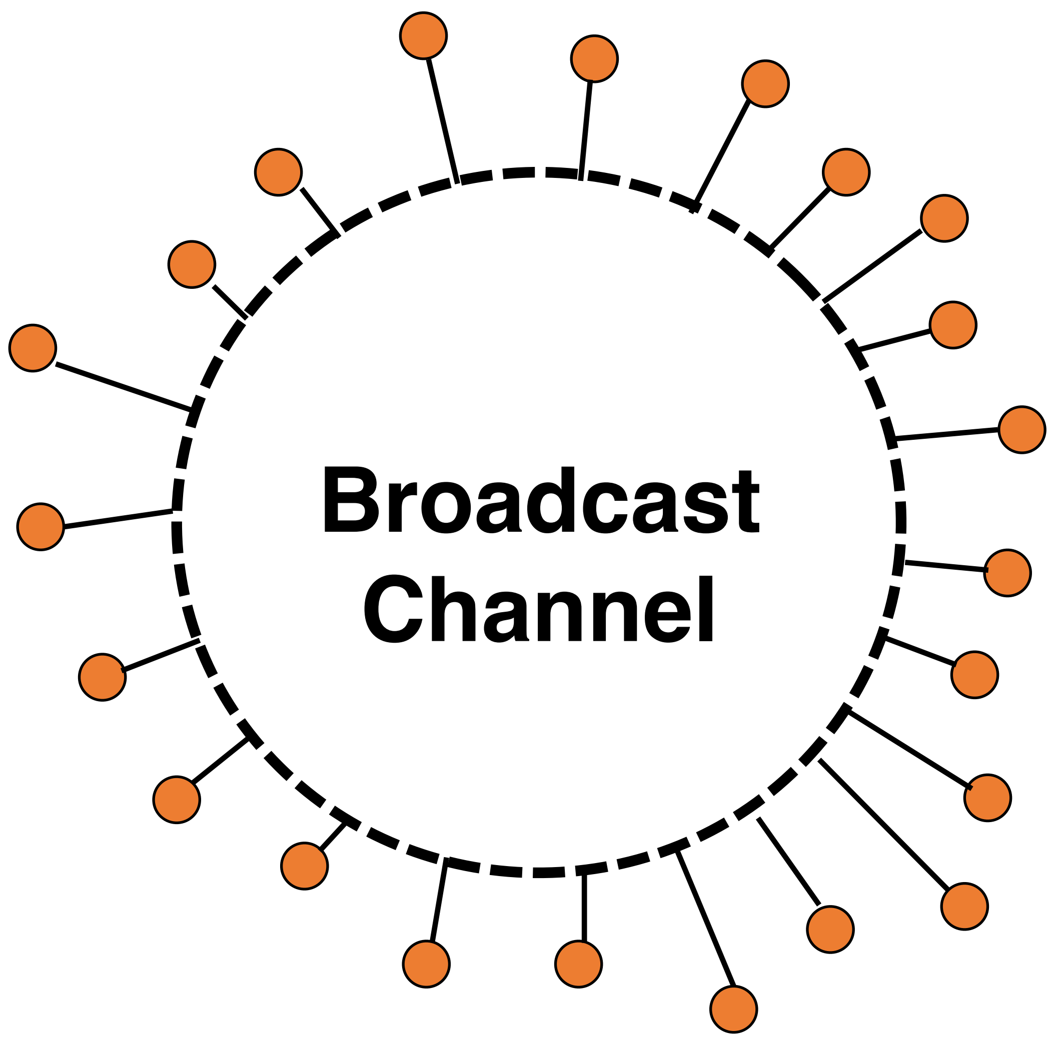


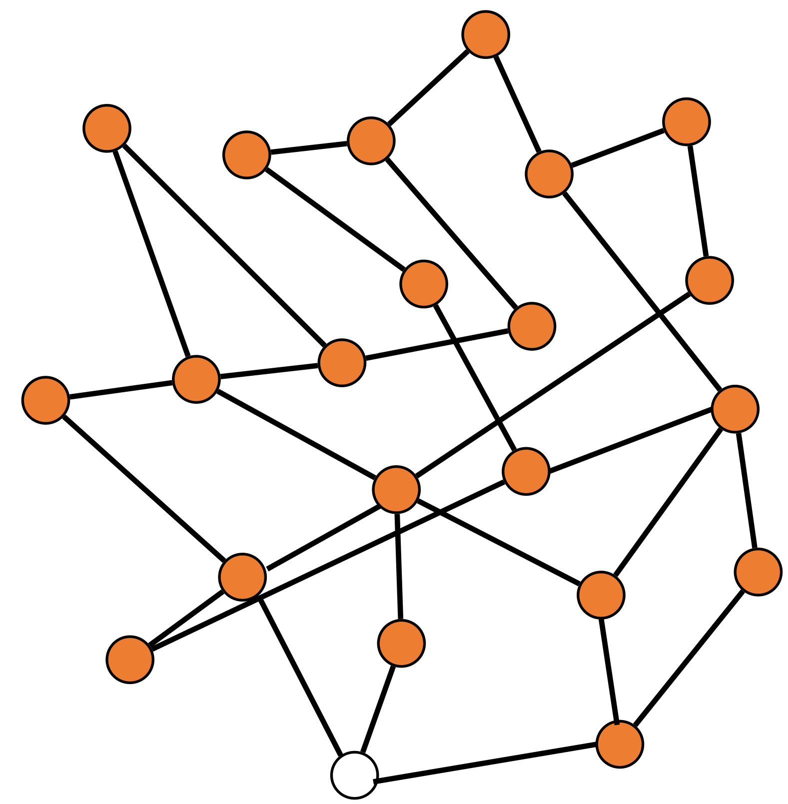
Broadcast
“Flooding”



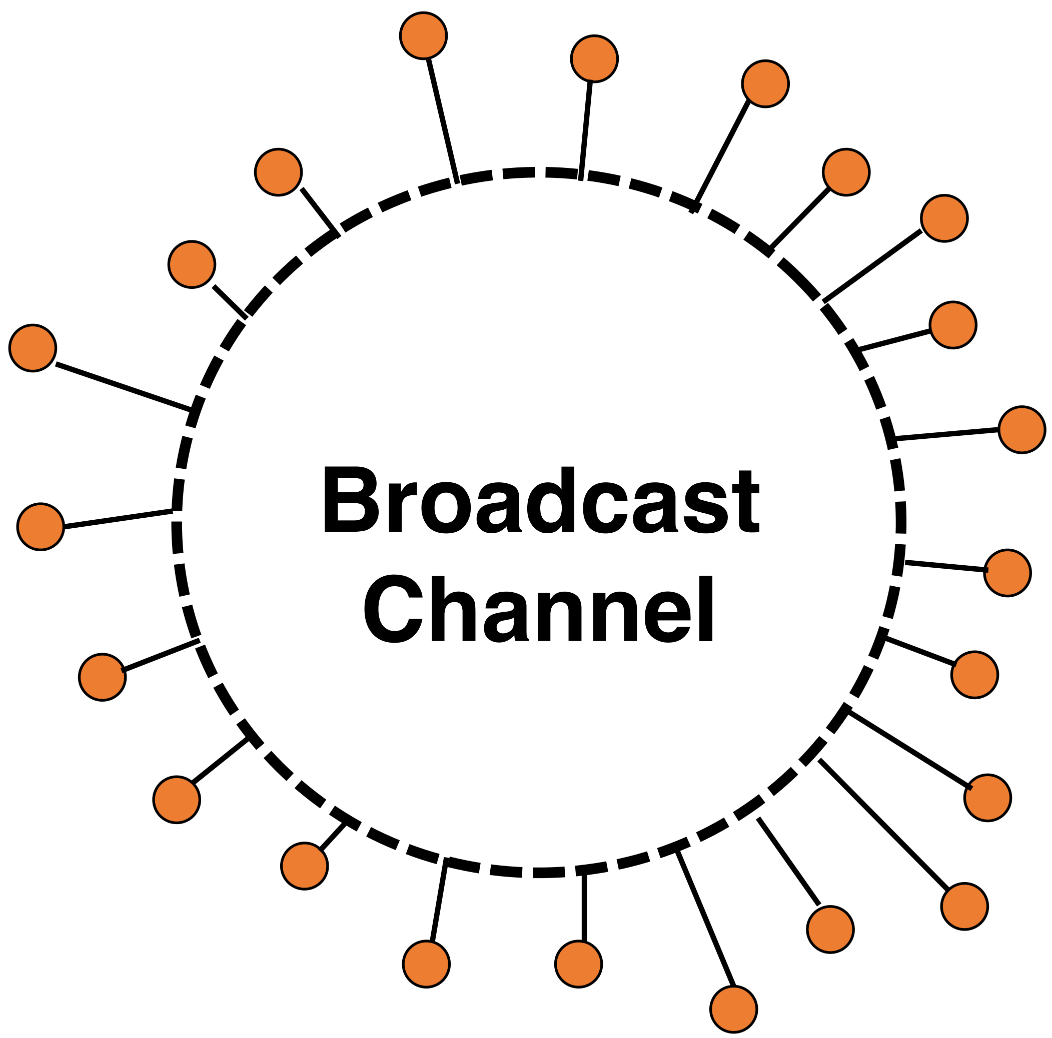


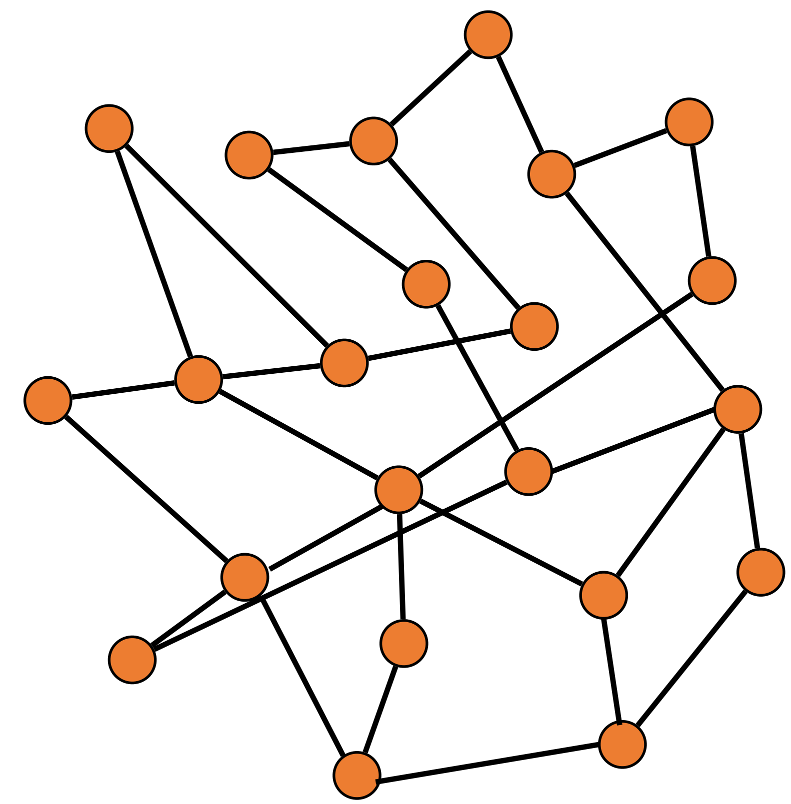
Broadcast
“Flooding”





Broadcast
“Flooding”





Broadcast
“Flooding”

Communication Complexi

for 1 Message is $\Theta(n)$.

Lightweight communication protocols:

- constant ***c*** messages are broadcast;
- communication complexity is $\Theta(n)$.
- can scale to a huge network

Heavy communication protocols:

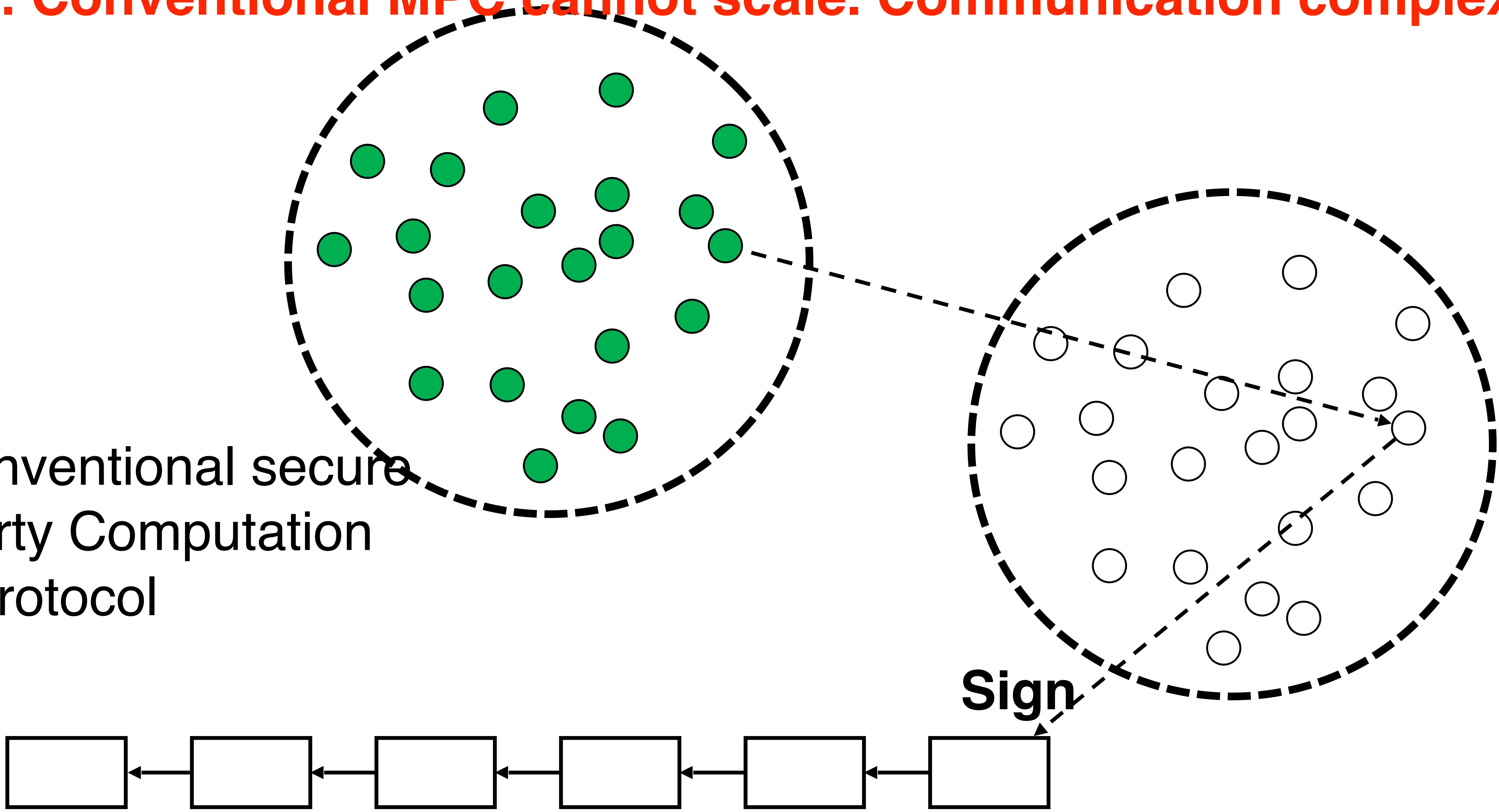
- such as voting needs;
- $\Theta(n)$ messages are broadcast;
- communication complexity is $\Theta(n^2)$.
- It is not scalable.

Conventional MPC-based Blockchain

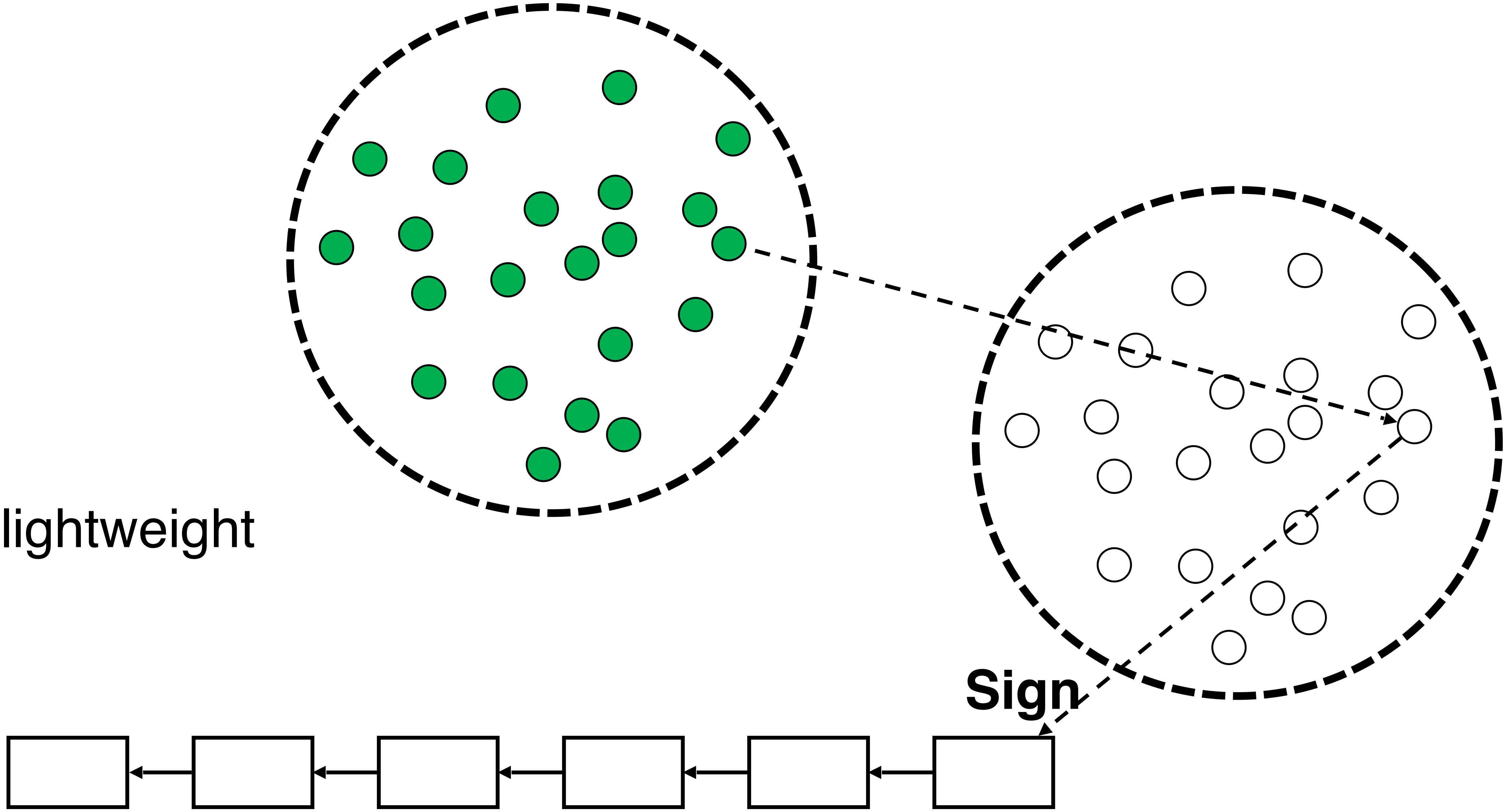
our goal is to obtain a large-scale blockchain.

Warning: Conventional MPC cannot scale. Communication complexity (n^2)

run a conventional secure Multi-Party Computation (MPC) protocol

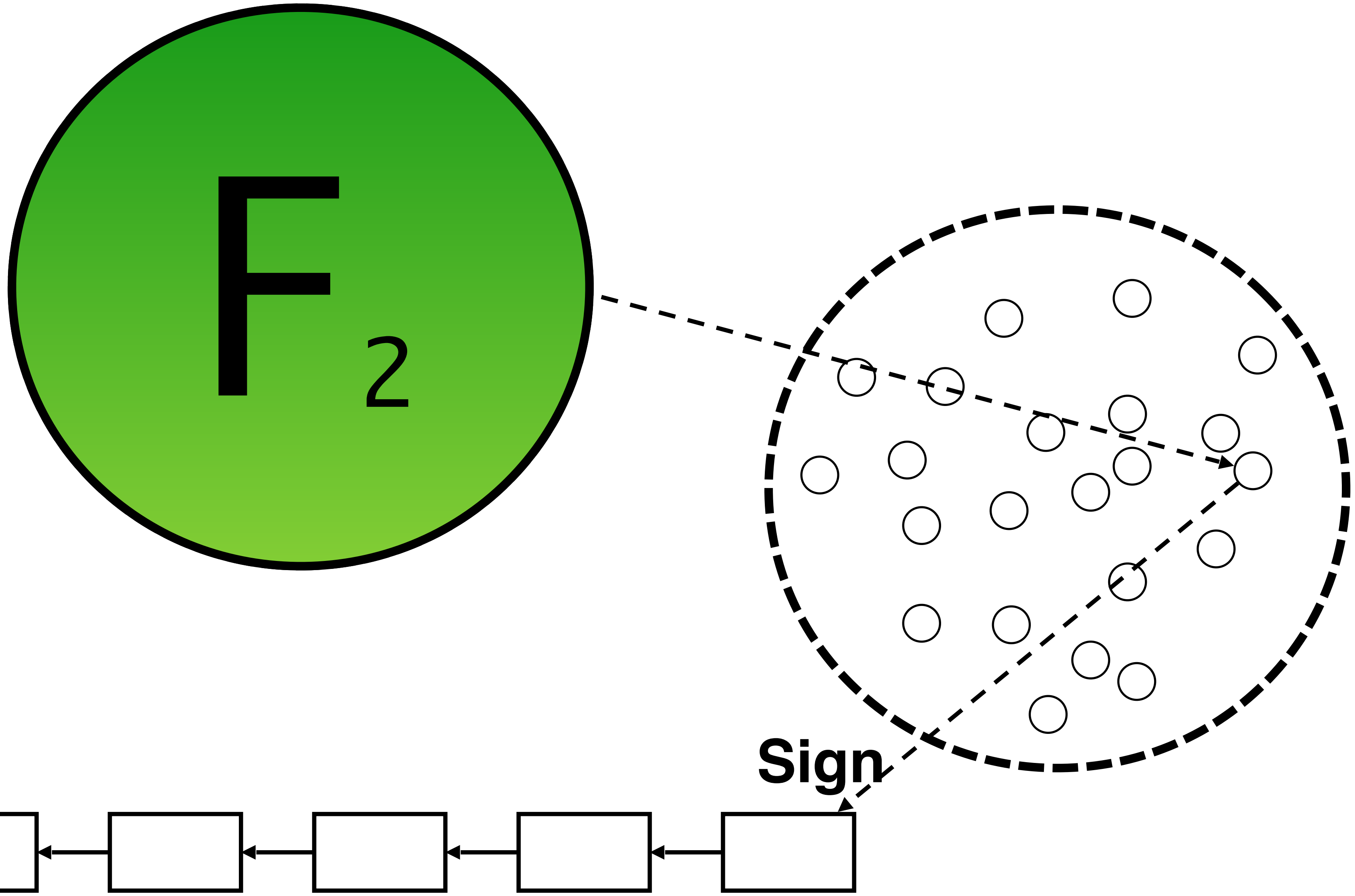


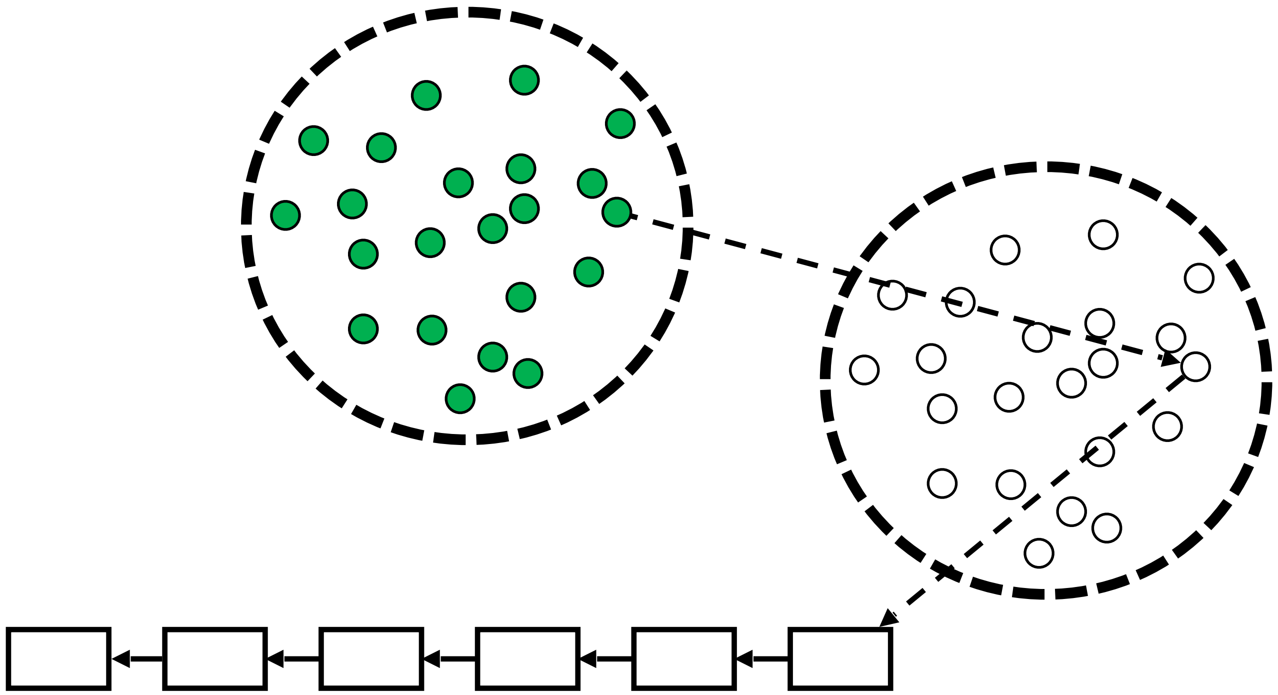
only lightweight blockchains scale



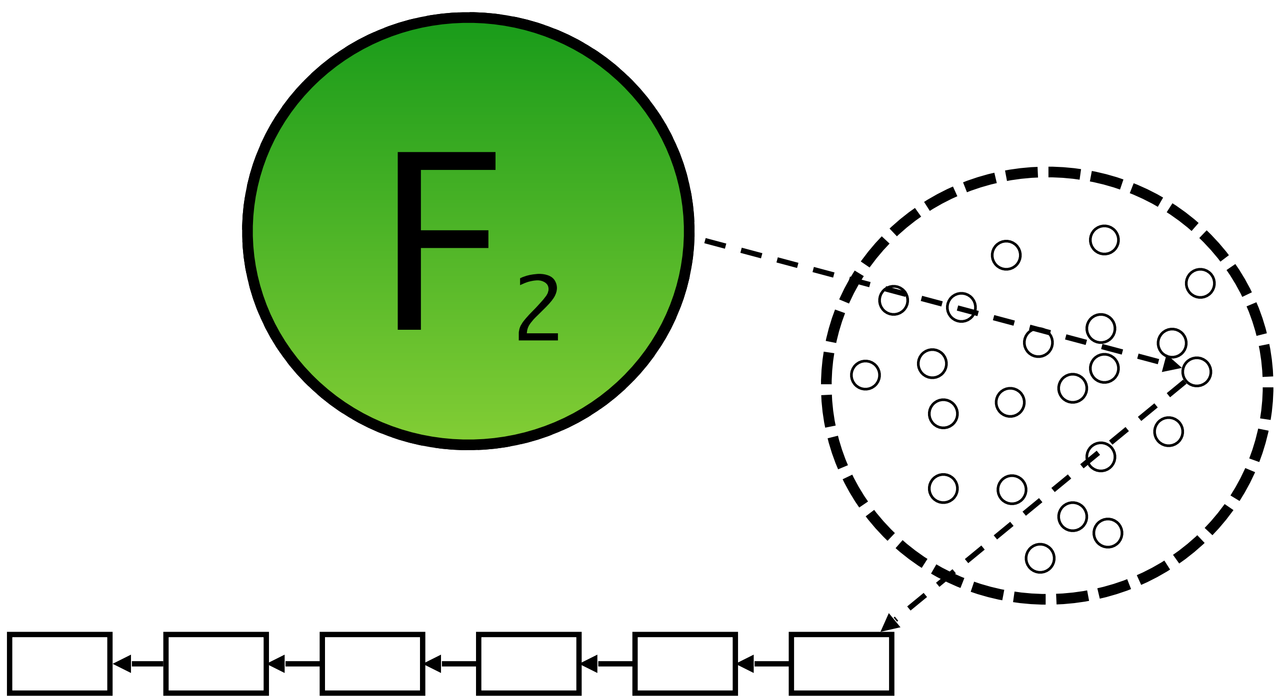
expect a lightweight protocol

consider an *environment friendly* beacon functionality





if we can design a lightweight protocol

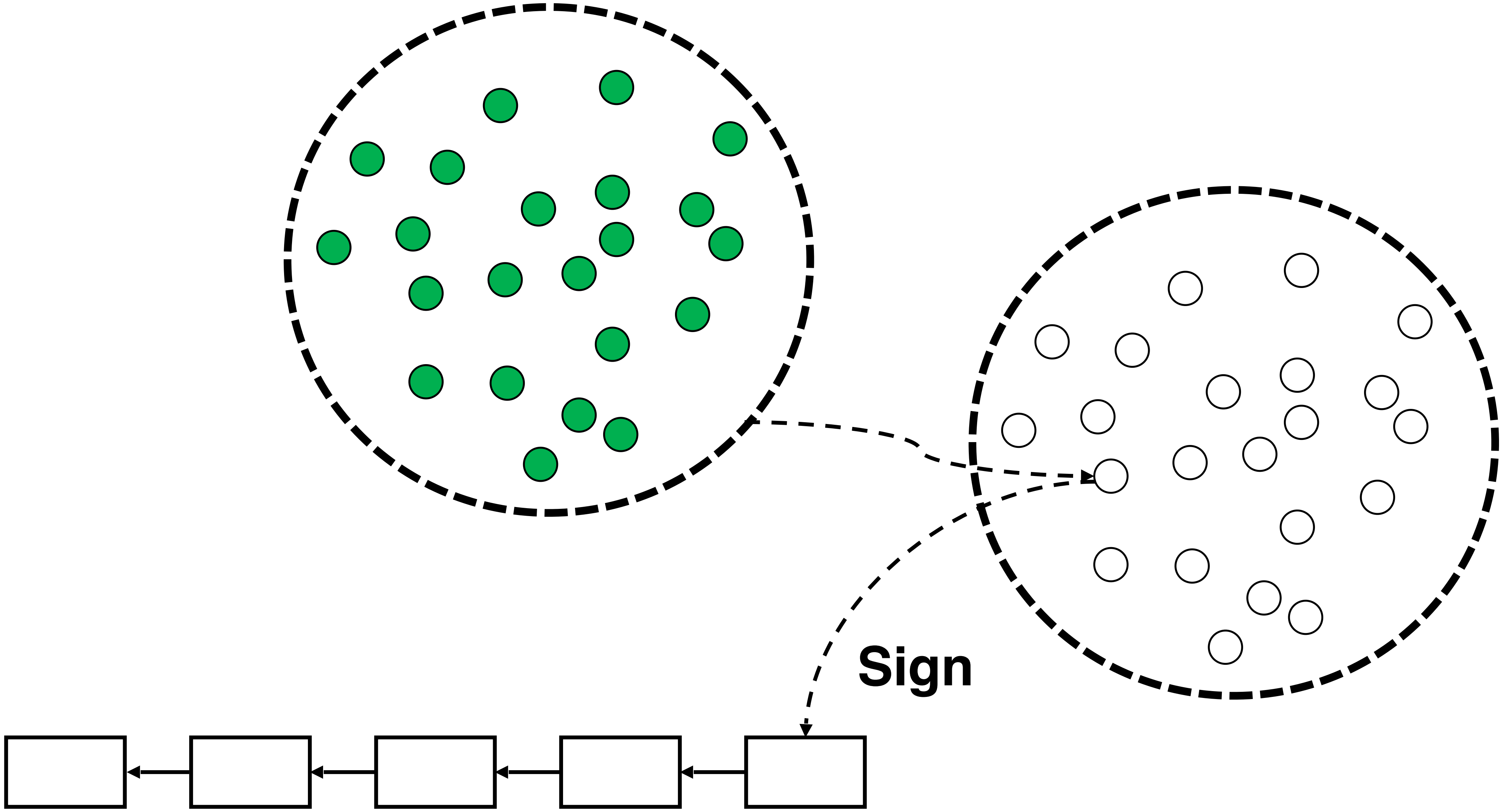


which achieves an *environment friendly* beacon functionality

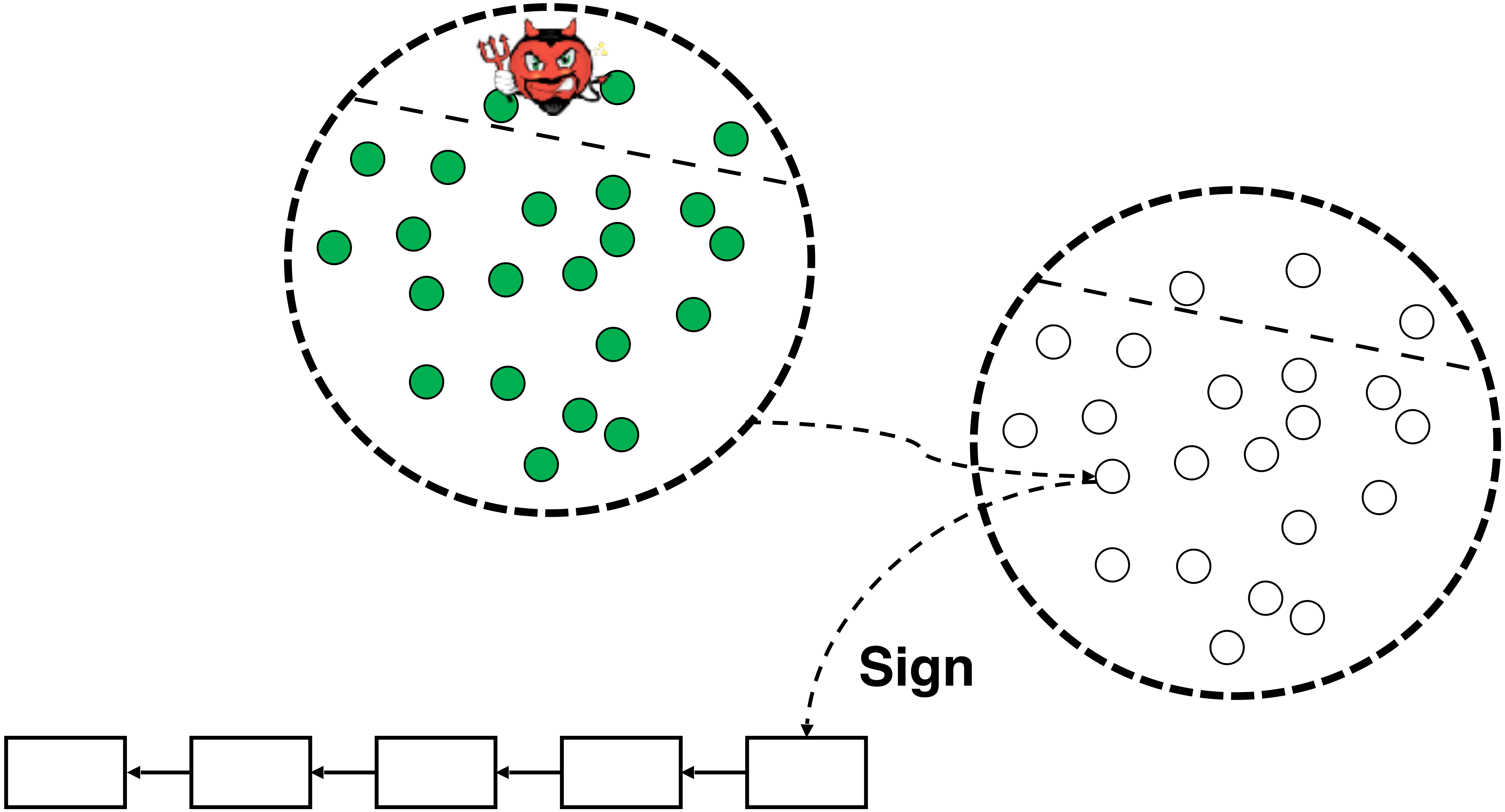
then we could make a better blockchain than Nakamoto's

However….main obstacle: splitting attack

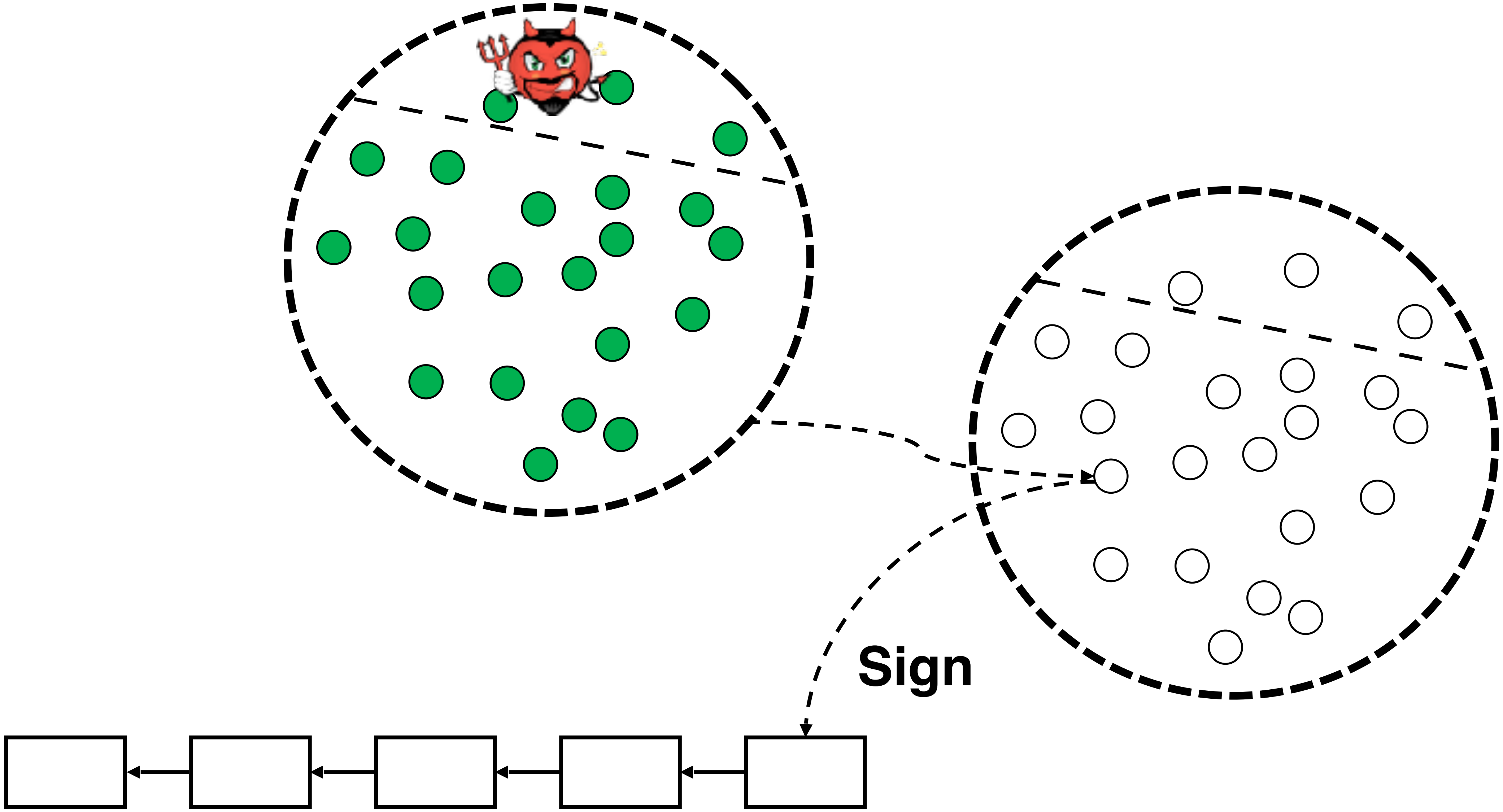
splitting attack on a class of lightweight protocols



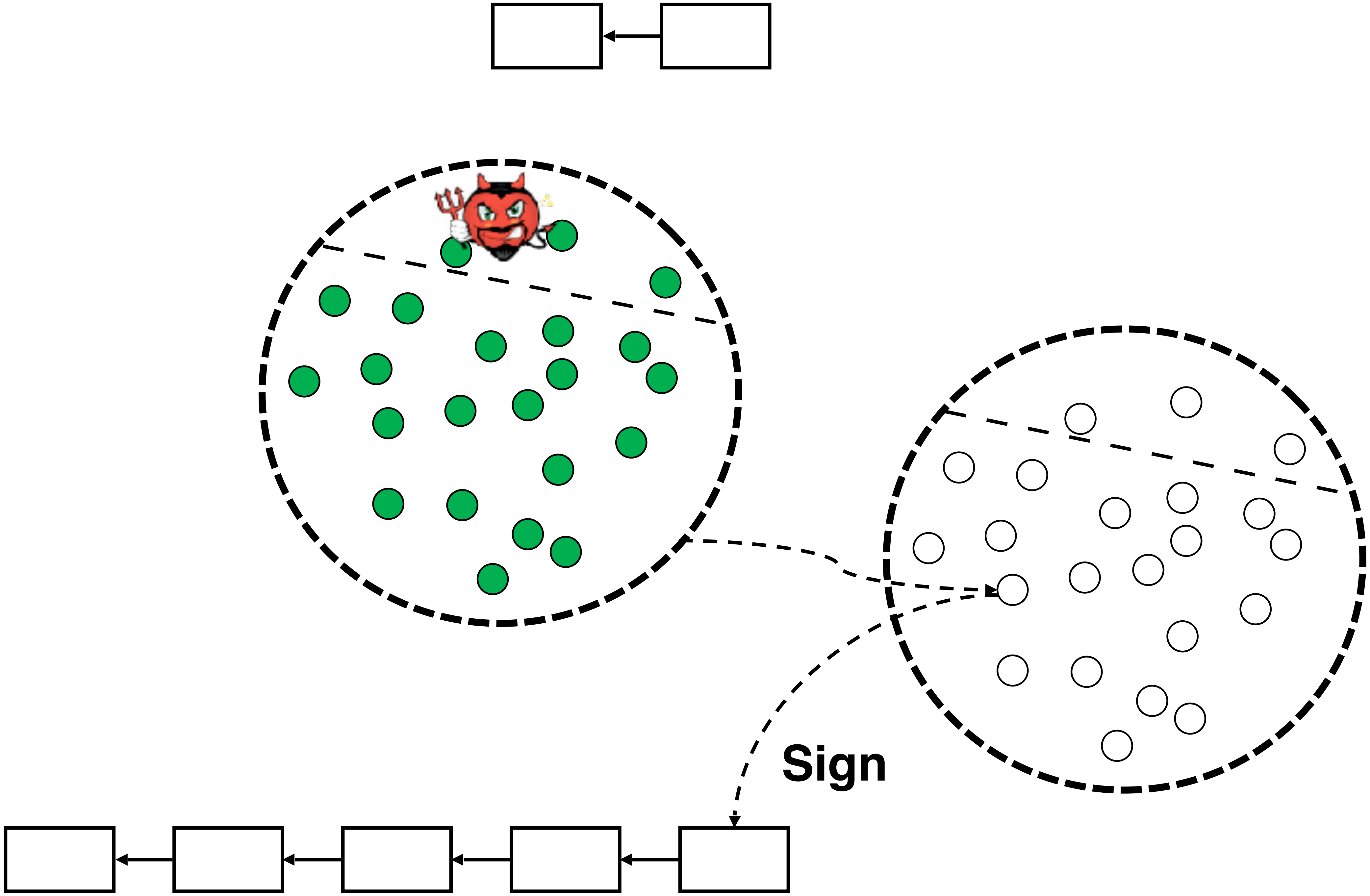
splitting attack on a class of lightweight protocols



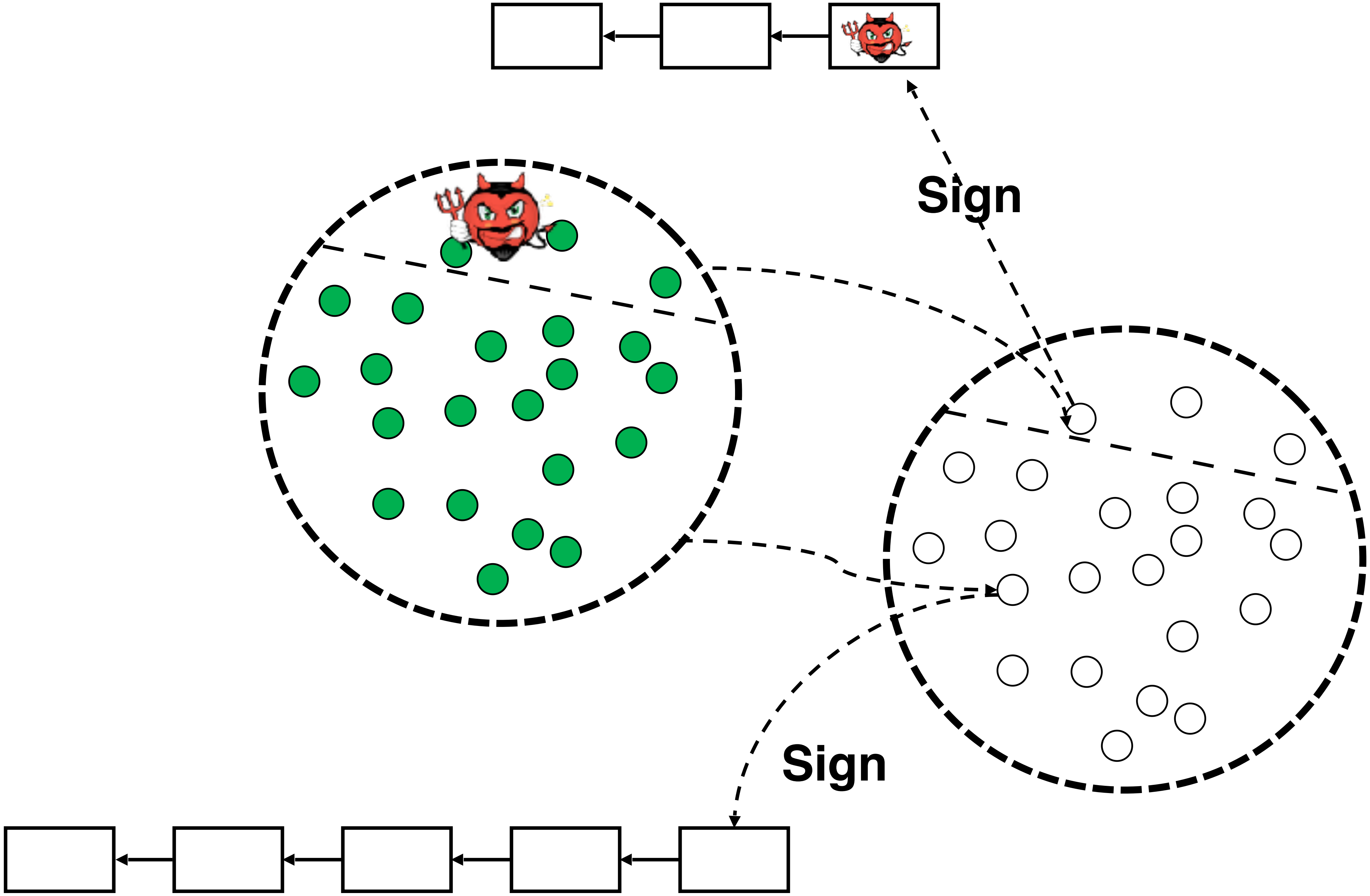
splitting attack on a class of lightweight protocols



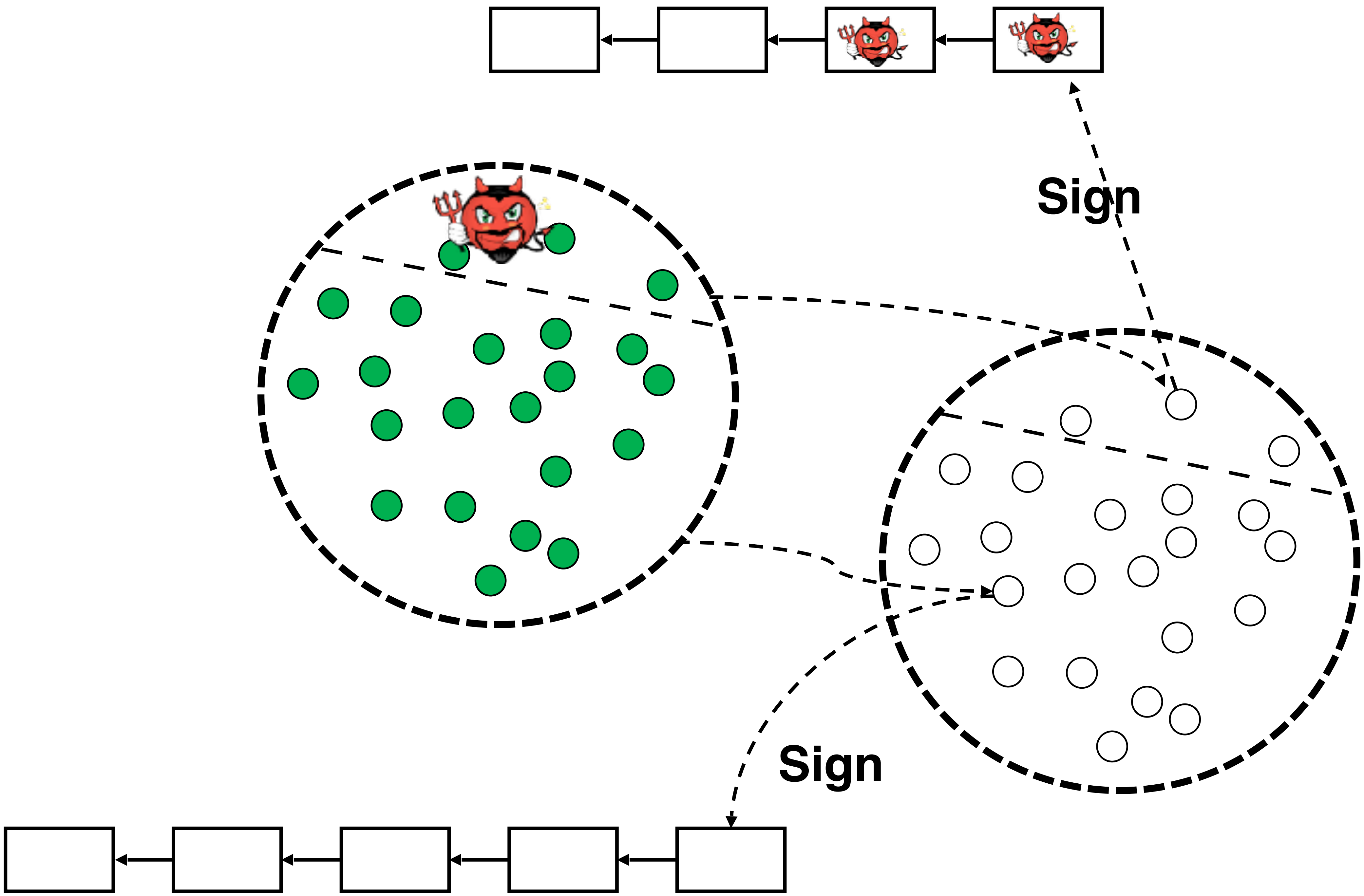
splitting attack on a class of lightweight protocols



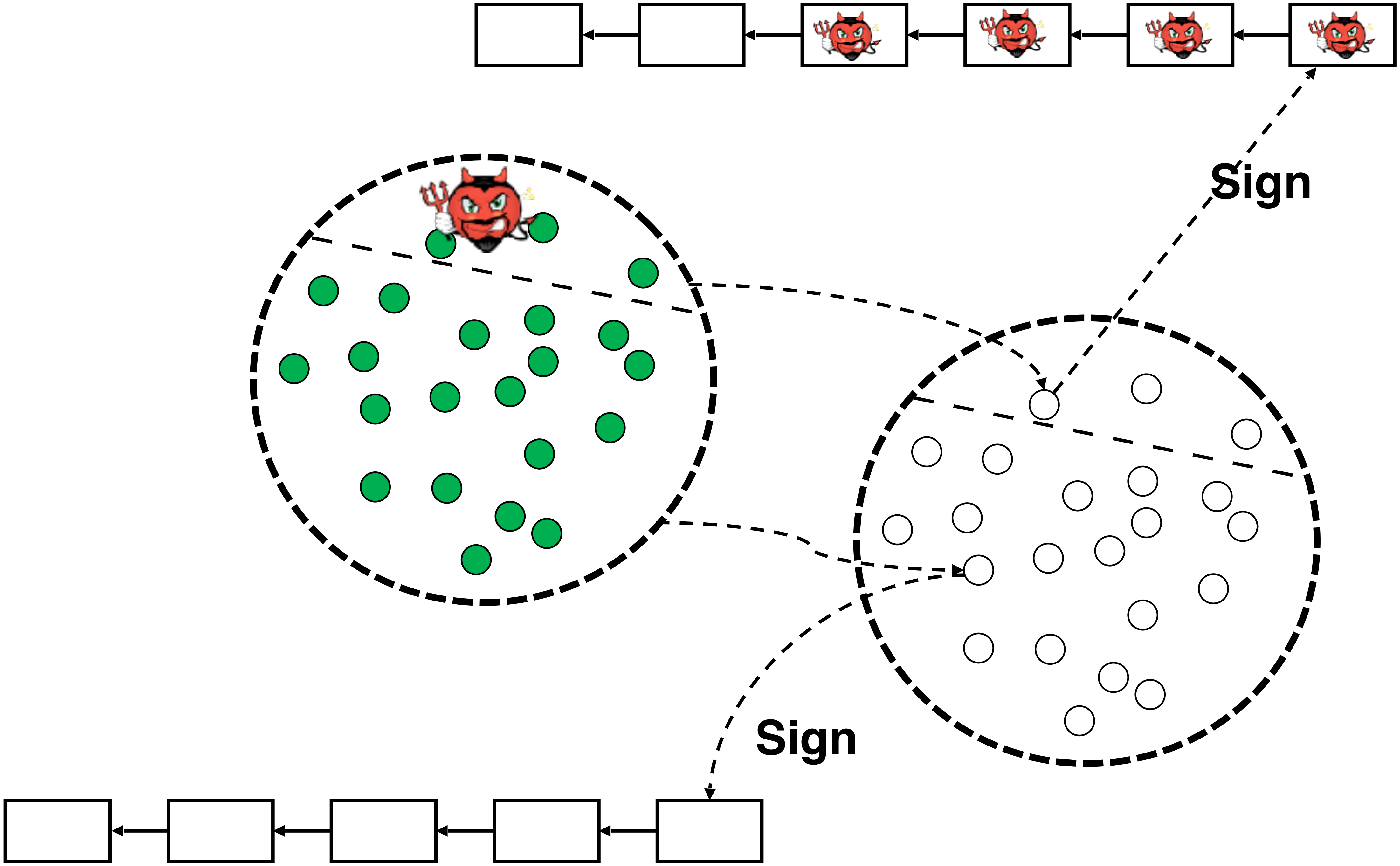
splitting attack on a class of lightweight protocols



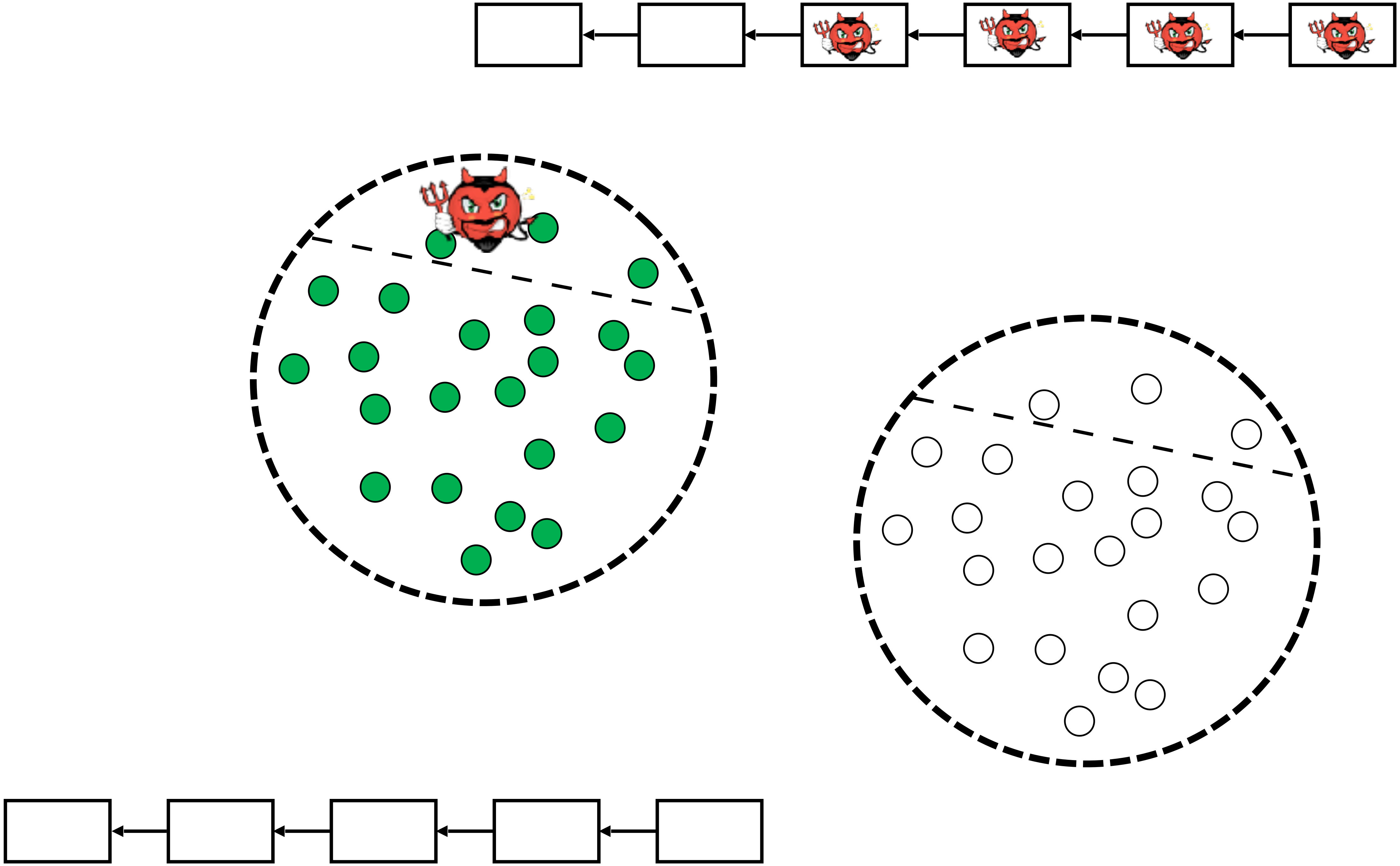
splitting attack on a class of lightweight protocols



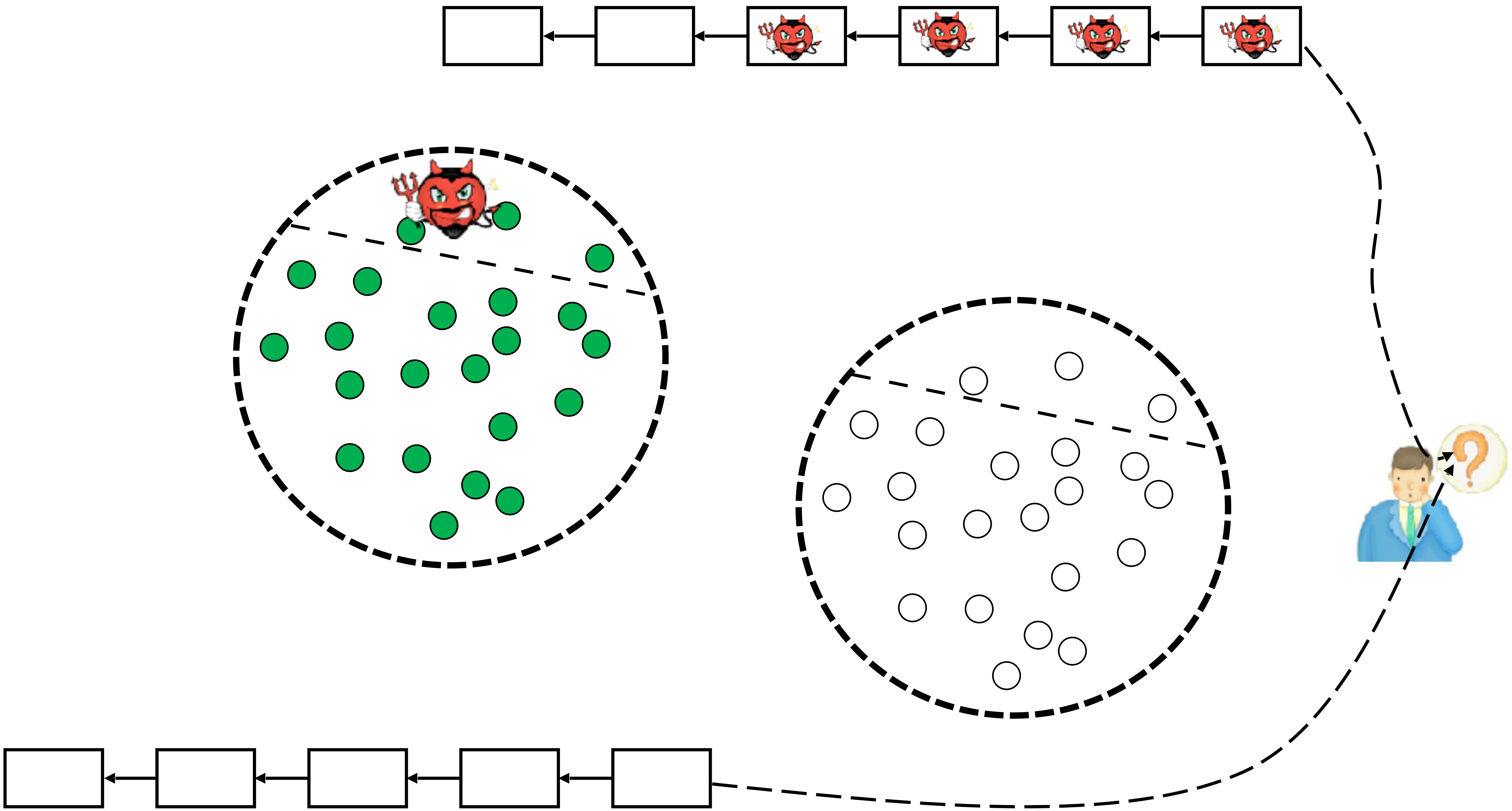
splitting attack on a class of lightweight protocols



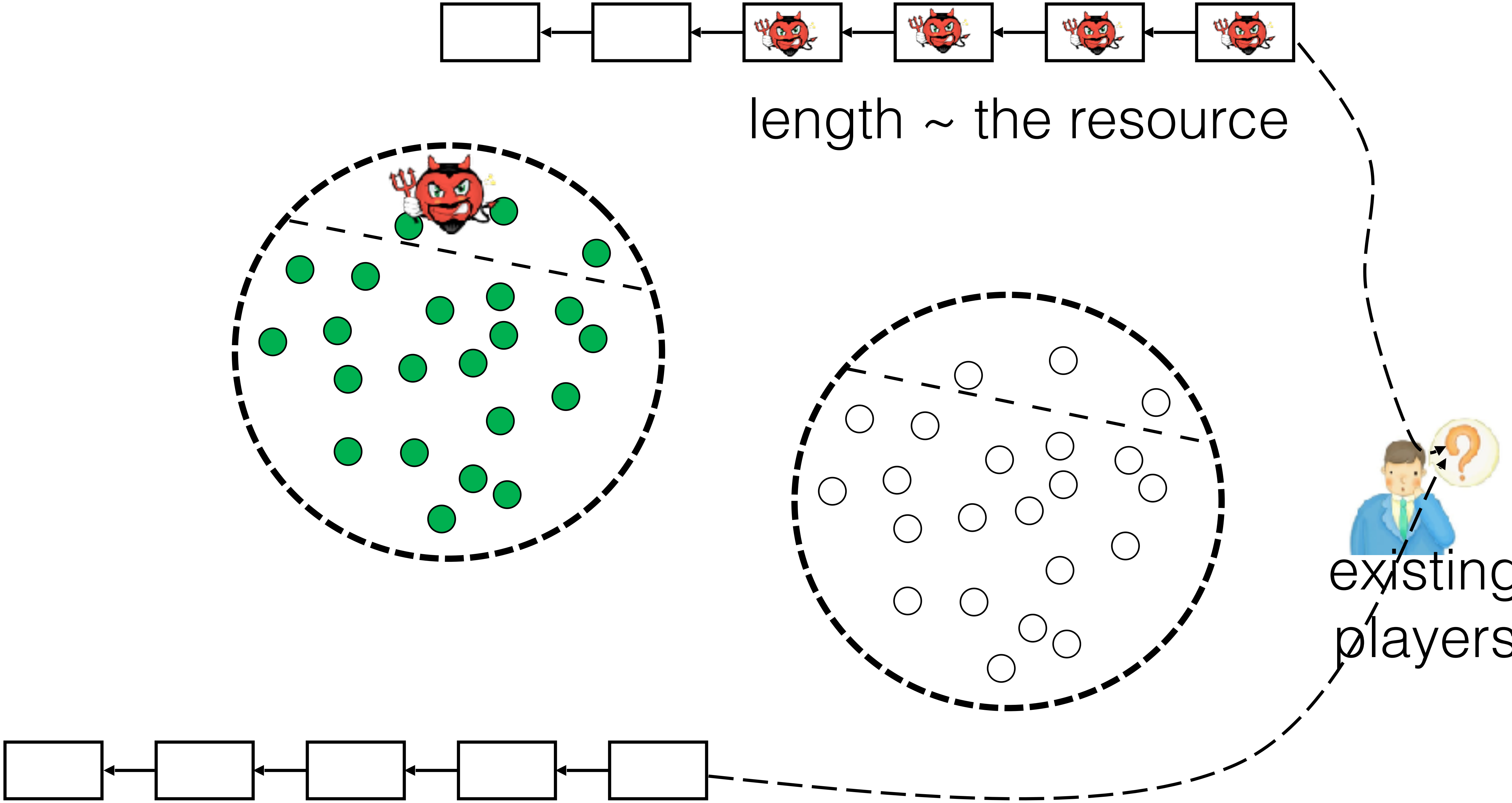
splitting attack on a class of lightweight protocols



splitting attack on a class of lightweight protocols

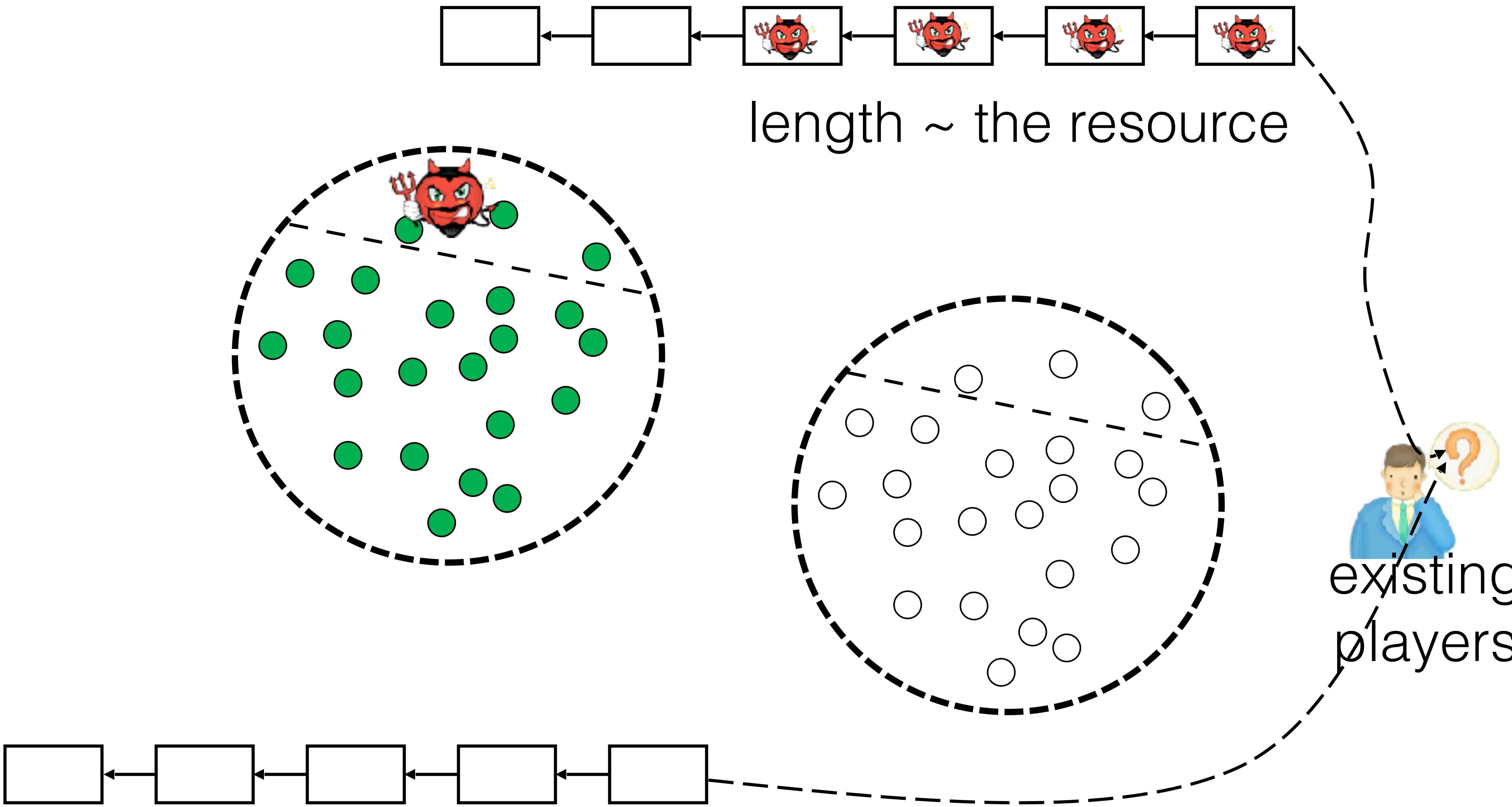


splitting attack on a class of lightweight protocols

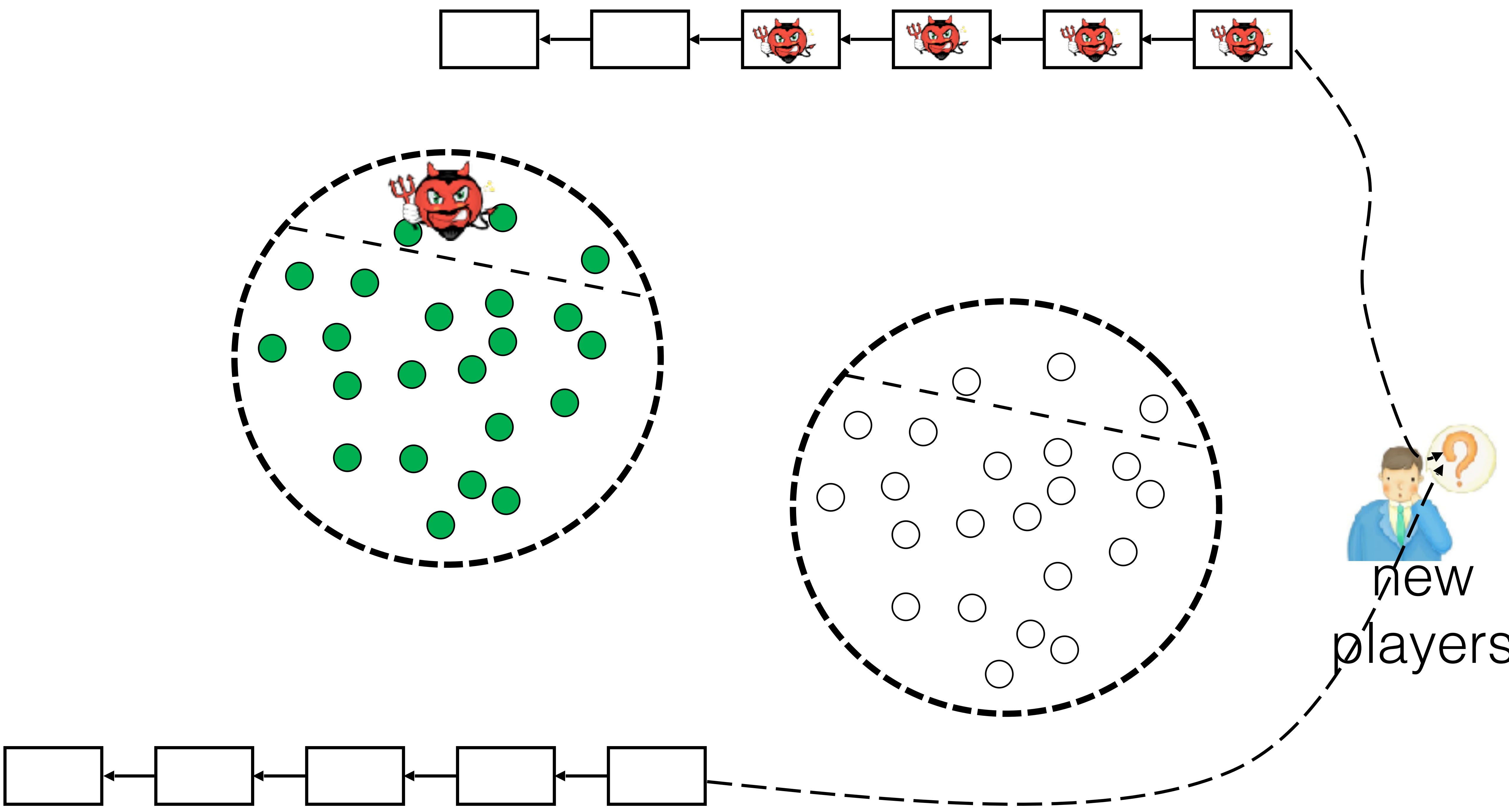


splitting attack on a class of lightweight protocols

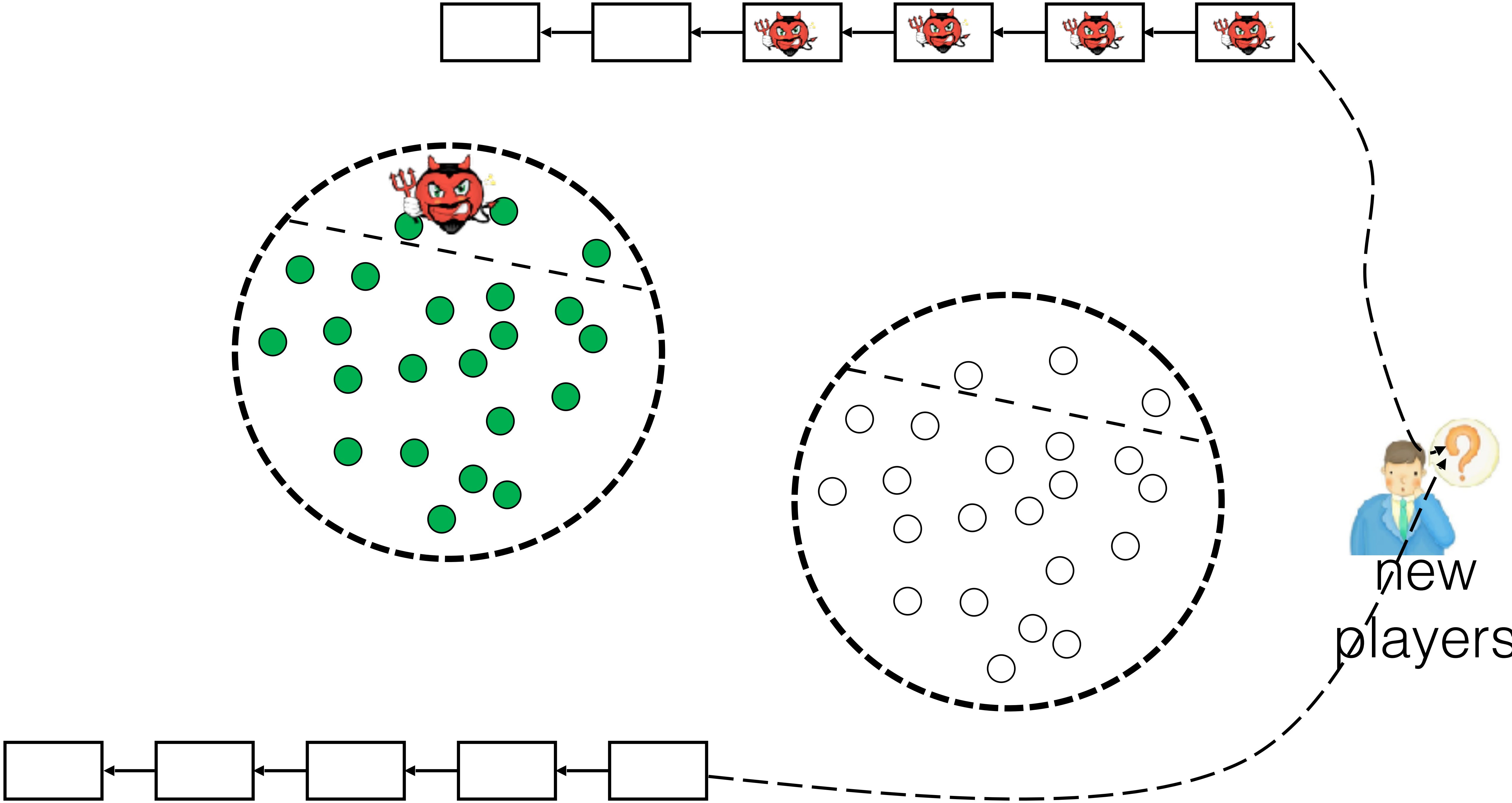
not a concern



splitting attack on a class of lightweight protocols



a big concern



However….main obstacle: splitting attack

Is that possible to fix the issue?

However....main obstacle:
splitting attack

Is that possible to fix the issue?
Yes. players run a voting.

However….main obstacle: splitting attack

Is that possible to fix the issue?
Yes. players run a voting.

Voting is a conventional MPC, which cannot scale to a large network of nodes.

However....main obstacle: splitting attack

Is that possible to fix the issue?

However....main obstacle: splitting attack

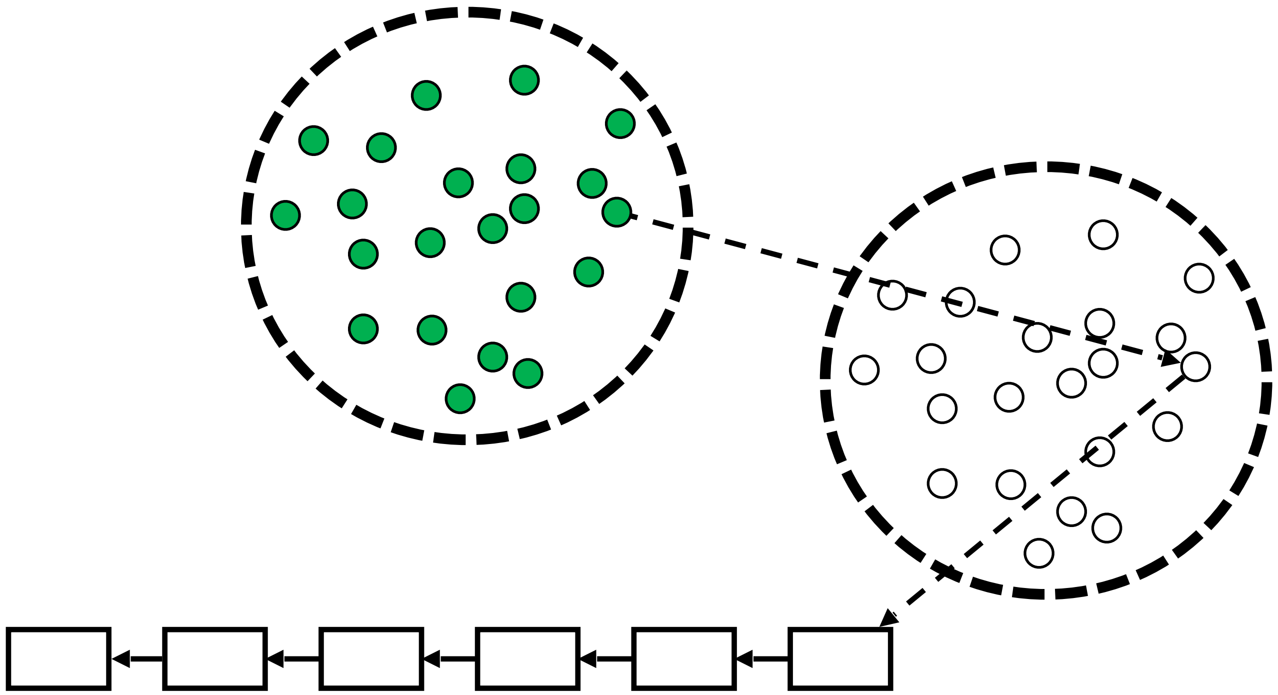
Is that possible to fix the issue?

Yes. via external checkpoints

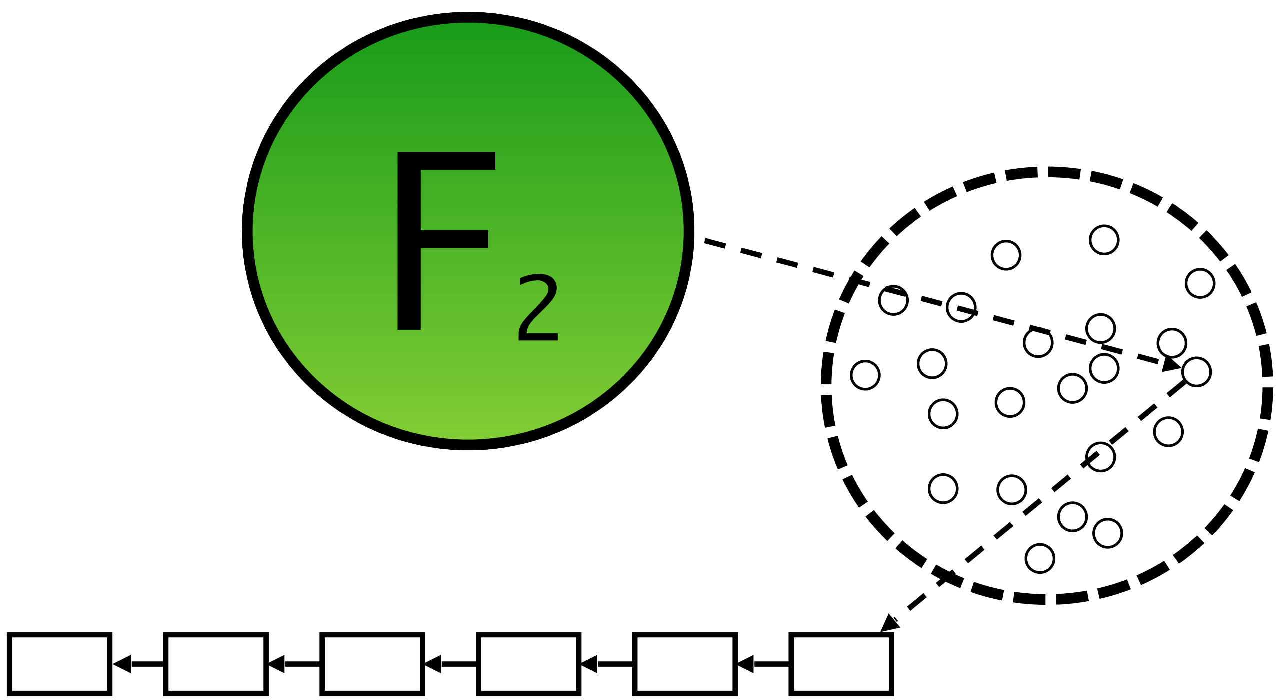
However....main obstacle:
splitting attack

Is that possible to fix the issue?
Yes. via external checkpoints

this violates the
decentralization.



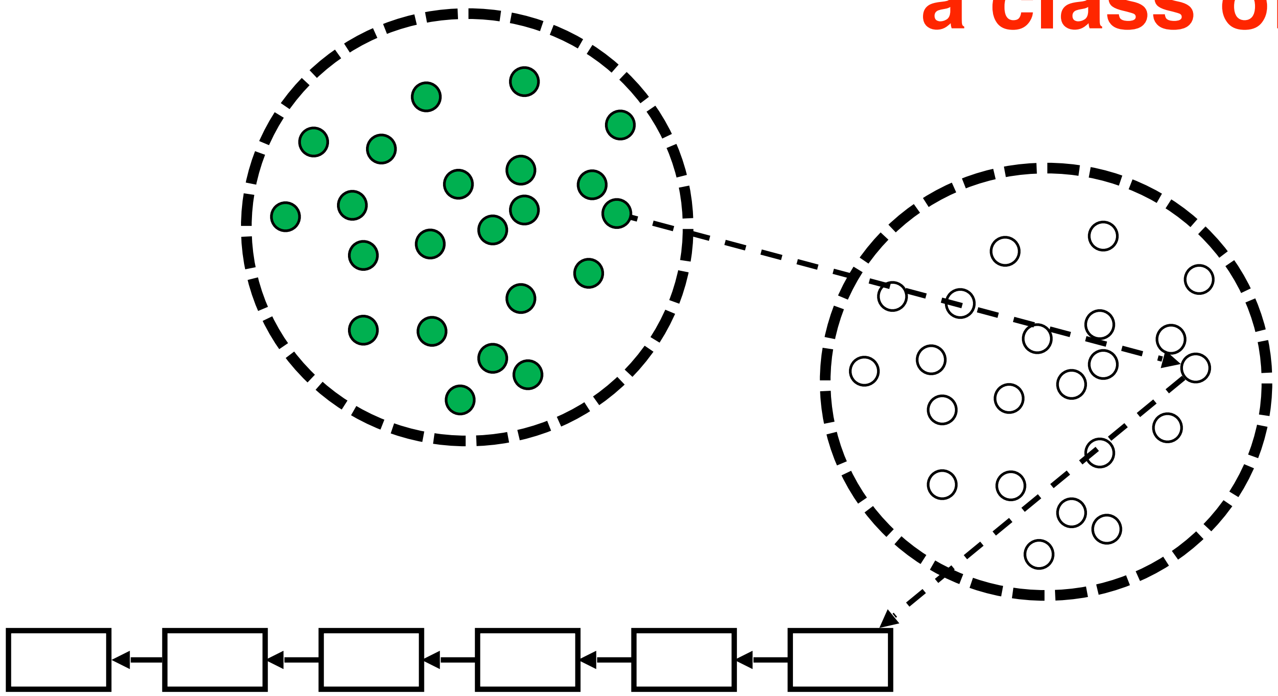
if we can design a lightweight protocol



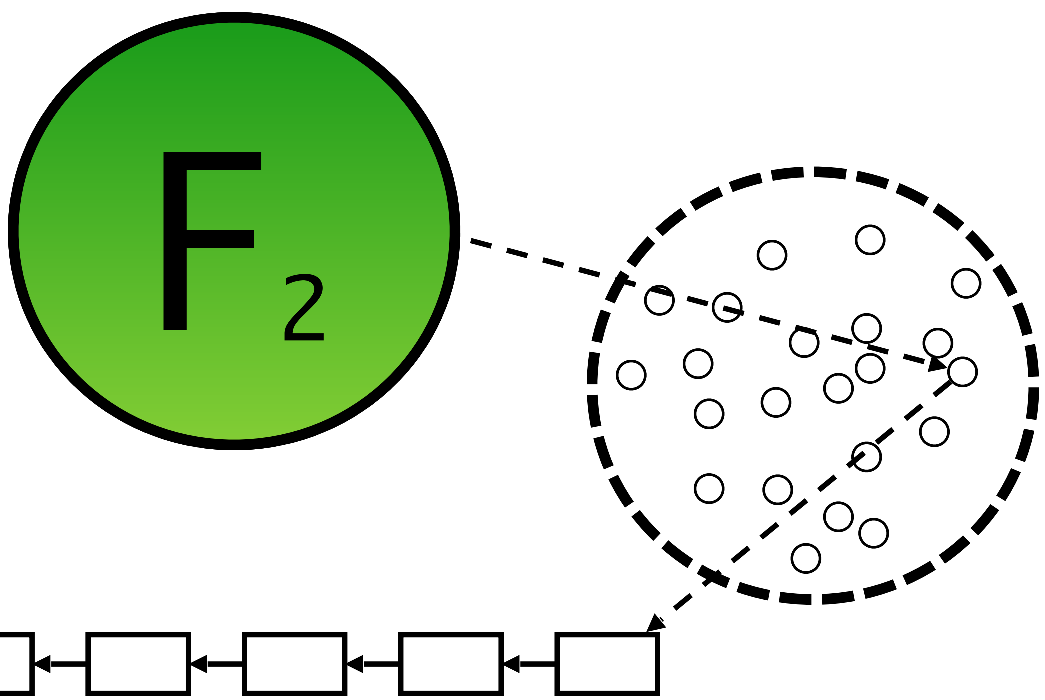
which achieves an *environment friendly* beacon functionality

then we could make a better blockchain than Nakamoto's

a class of lightweight protocols DO NOT



~~if we can design a lightweight protocol~~



which achieves an *environment friendly* beacon functionality

then we could make a better blockchain than Nakamoto's

Open Question

- Proof-of-stake blockchain
  - open
  - Internet-scale
  - provably secure

Interesting Question

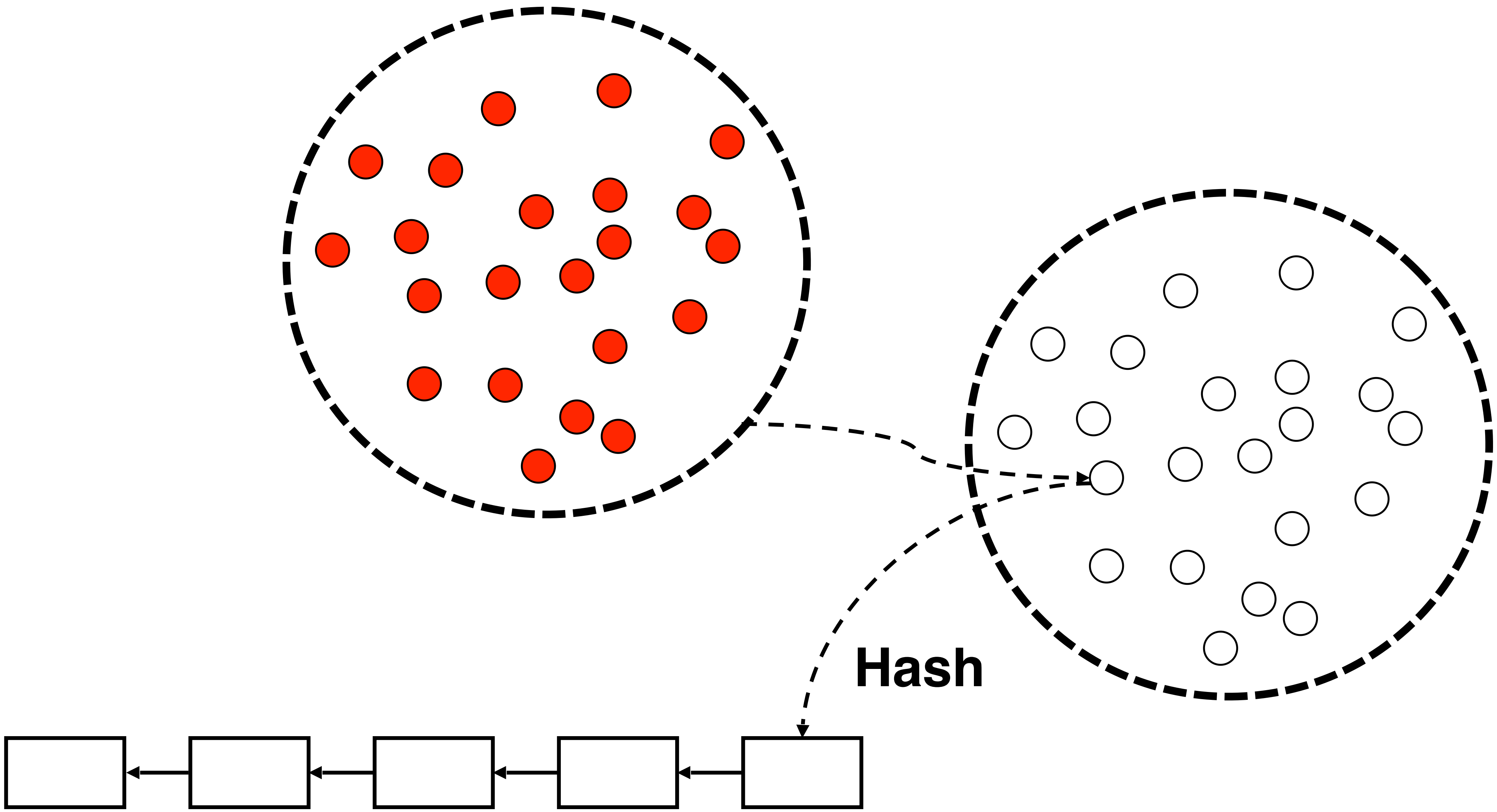
- Proof-of-stake blockchain
  - open
  - Internet-scale
  - provably secure

Stay tuned…. Lei's talk on Thursday

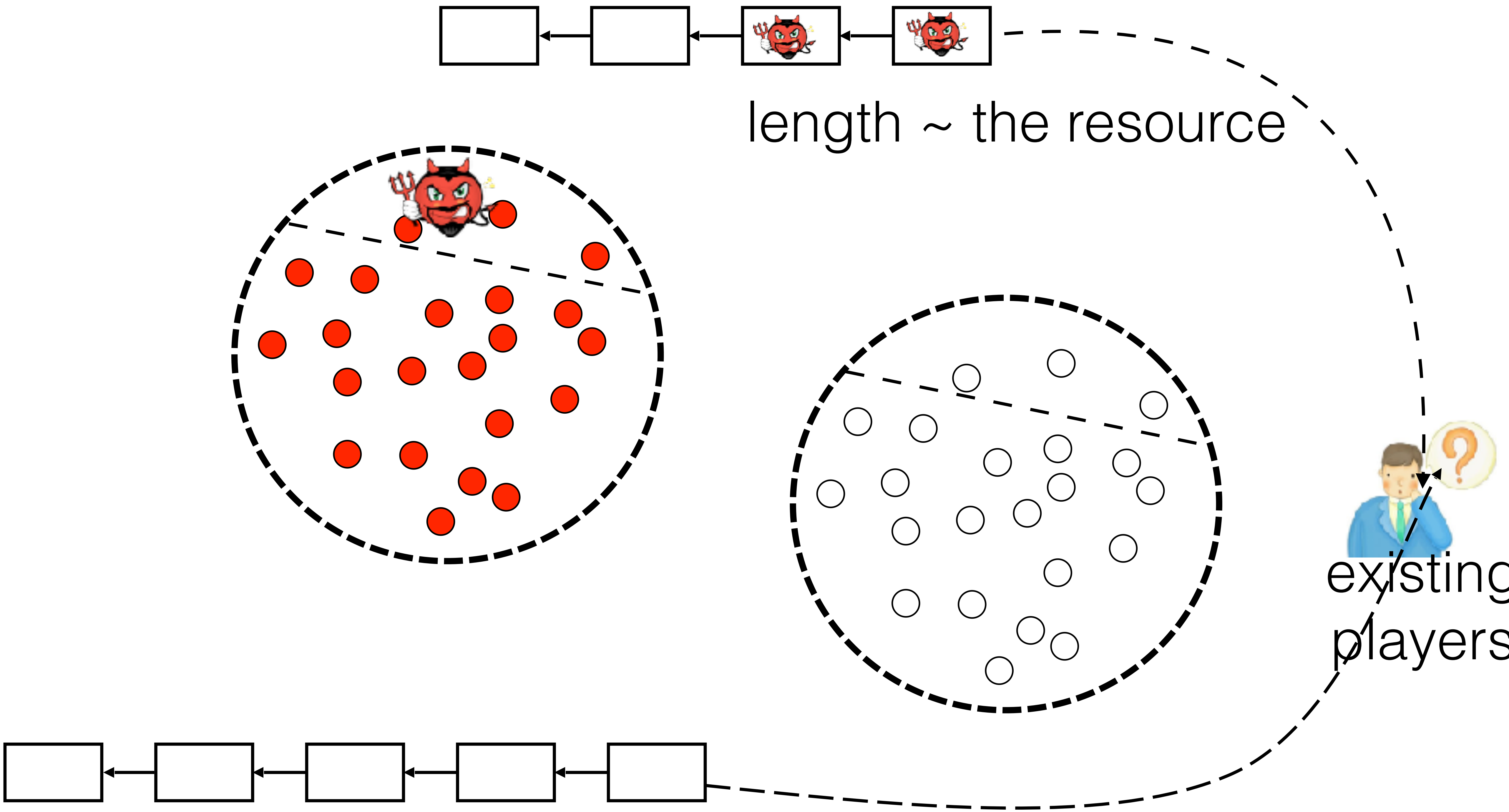
However....main obstacle: splitting attack

Is Nakamoto's design OK?

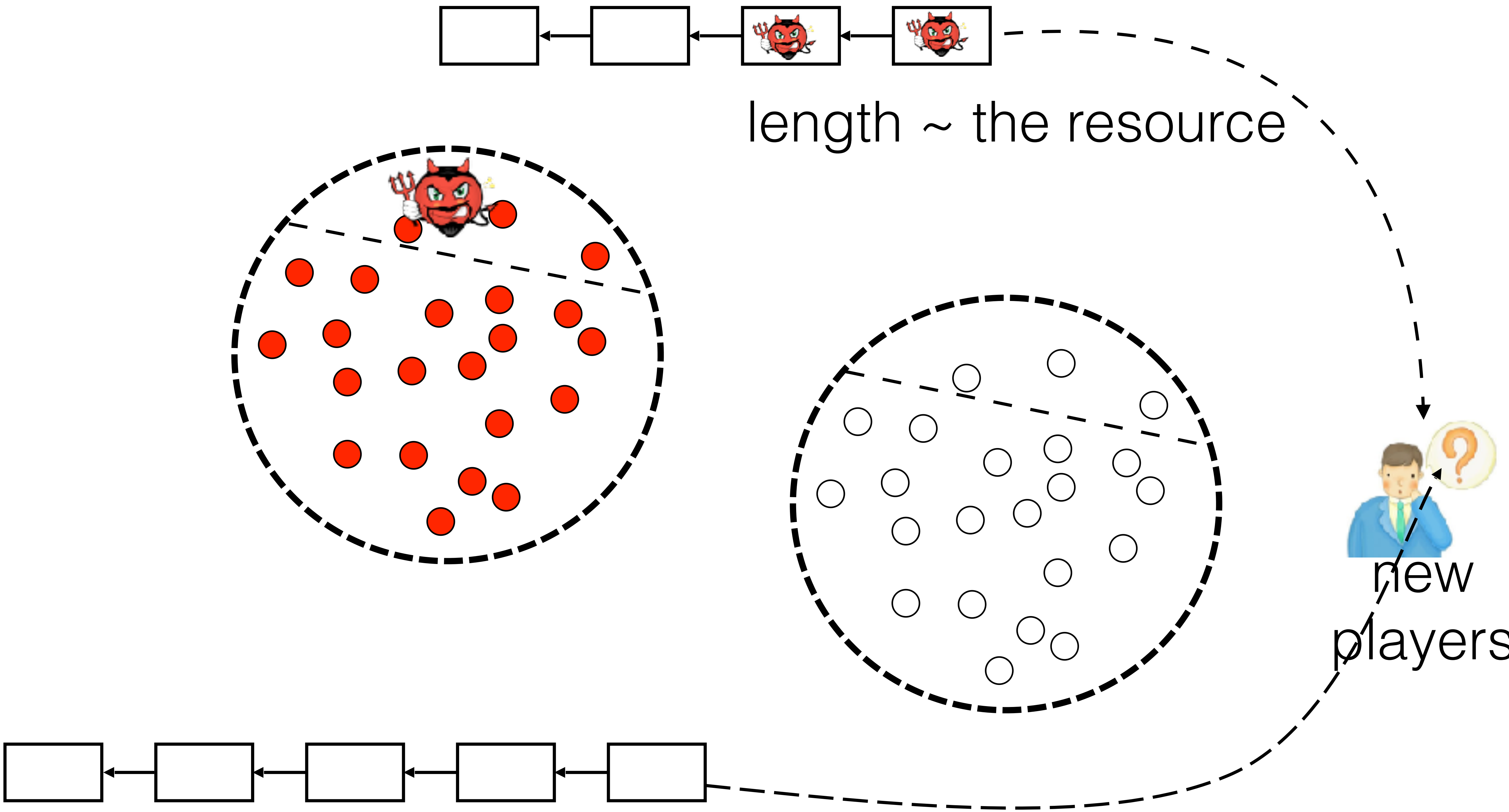
splitting attack on Nakamoto's design?

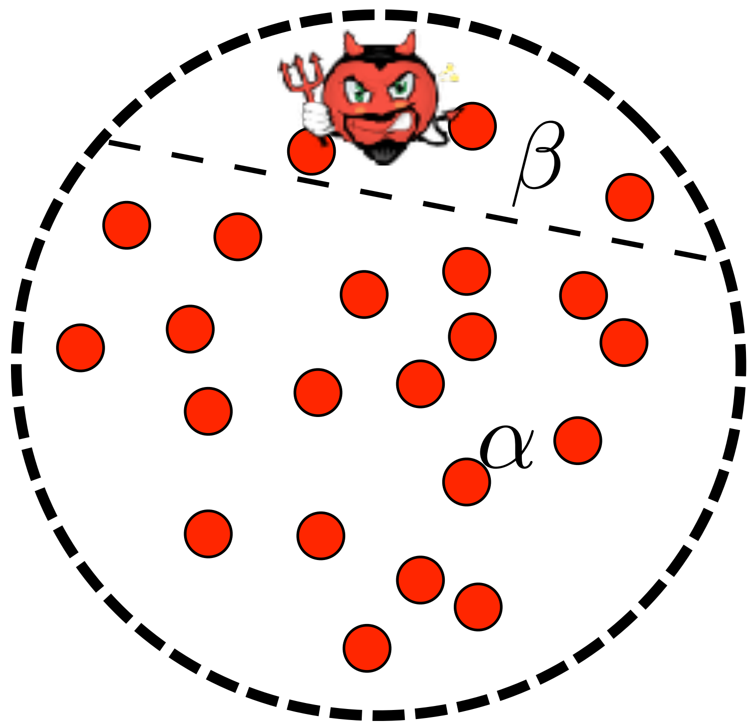


splitting attack on Nakamoto’s design?

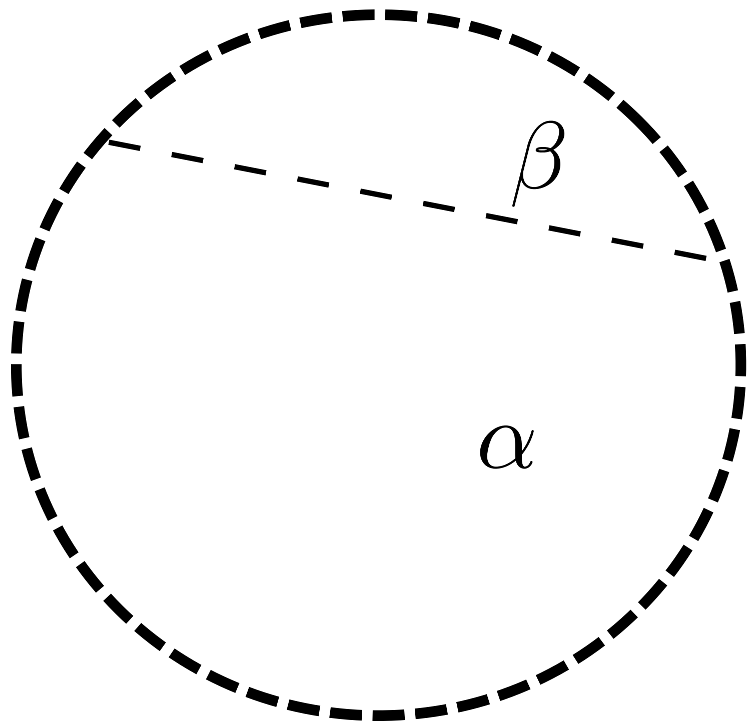


splitting attack on Nakamoto's design?





β
α


β
α

However....main obstacle: splitting attack

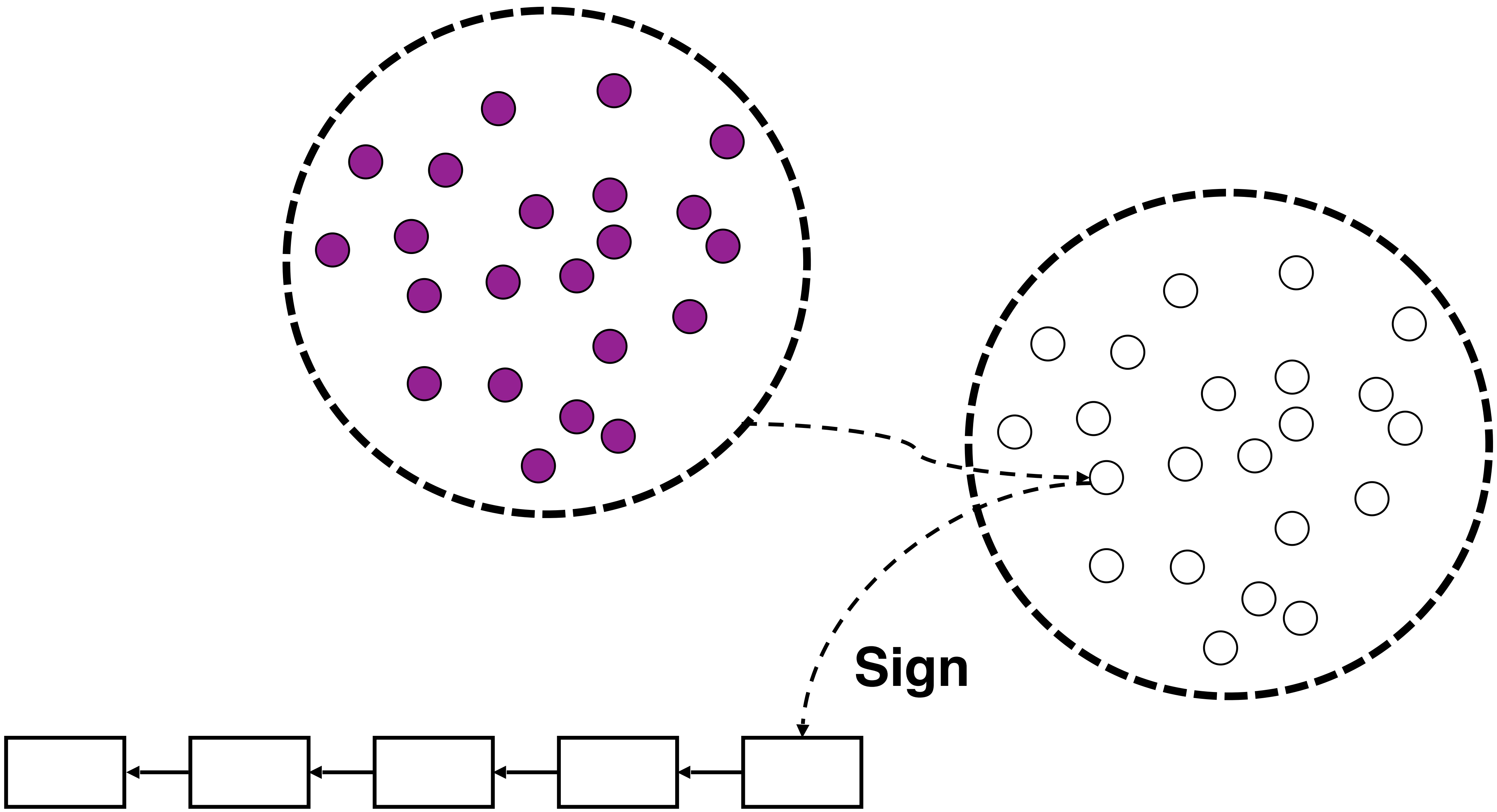
Is Nakamoto's design OK?

Yes.

However....main obstacle: splitting attack

Any other solutions against splitting attack?

trusted hardware based blockchains



However….main obstacle: splitting attack

Is this hardware based solution good?

However....main obstacle: splitting attack

Is this hardware based solution good?

No. Trapdoor available to a single party

Open Question

- hardware-based blockchain
  - trapdoor-resilient

However....main obstacle: splitting attack

Any other solutions against splitting attack?

Yes. Proof of X

X={Work, Storage, ... Human-work, ...}

However….main obstacle: splitting attack

Any other solutions against splitting attack?

Yes. Proof of X

X={Work, Storage, … Human-work, …}

useful work, combining work with storage, memory hard PoW

However....main obstacle: splitting attack

Any other solutions against splitting attack?

Yes. Proof of X

X={Work, Storage, … Human-work, …}

useful work, combining work with storage, memory hard PoW

Are they good?

References

- Modeling idea:
  Garay, Kiayias, Zhou, CSF 10
- Proof-of-Stake:
  Orborous; Snow White;
- Proof-of-X:
  PoET; SpaceMint; PermaCoin; PrimeCoin; PoST; memory hard PoW;

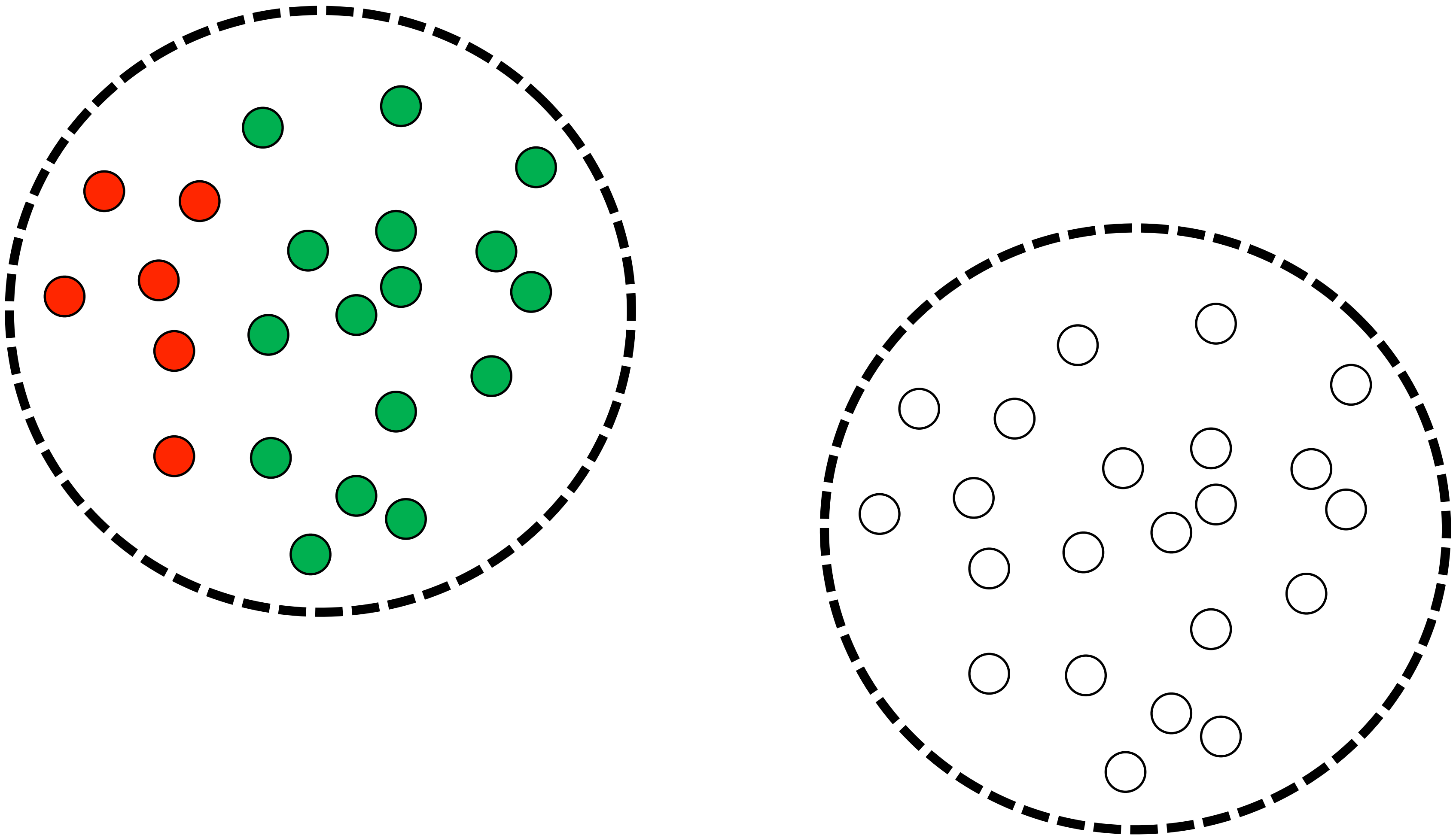
Alternative Mechanisms

A Design Example: 2-hop Blockchain

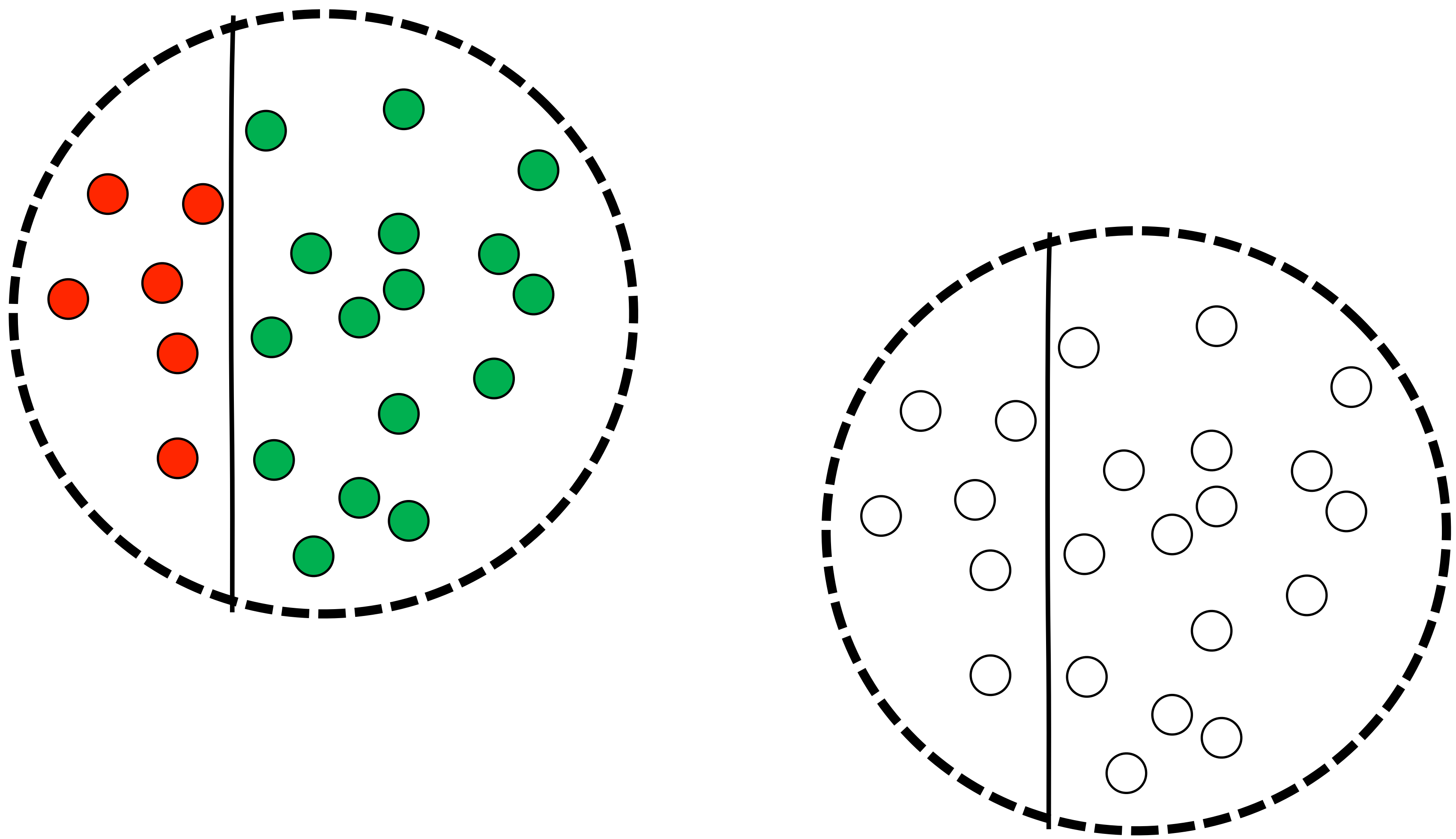
so far

- a unified view for constructing (a class of) open blockchains has been developed
- existing proof-of-stake based open blockchains **cannot** scale to a large number of nodes

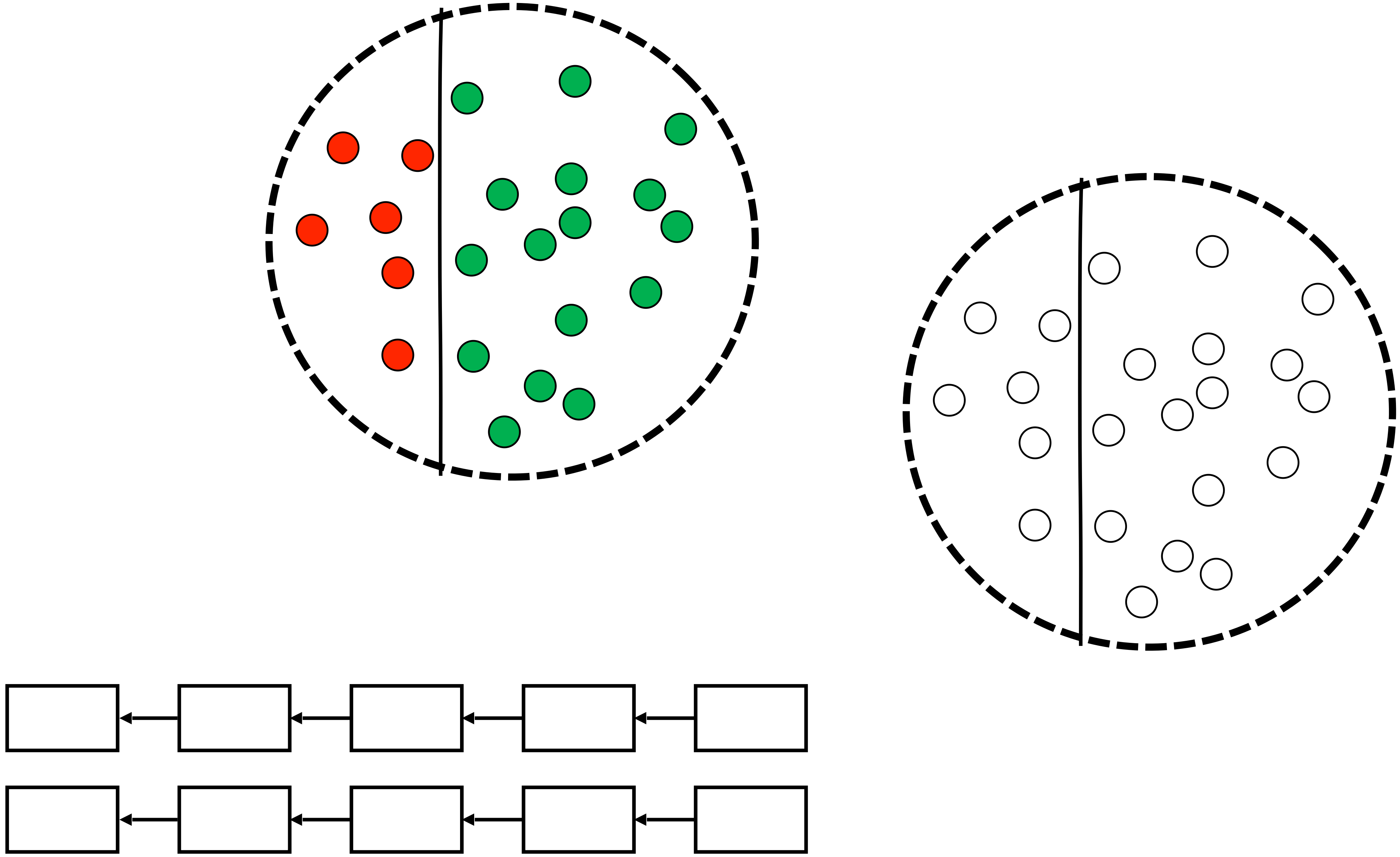
PoW/PoS-based Blockchain



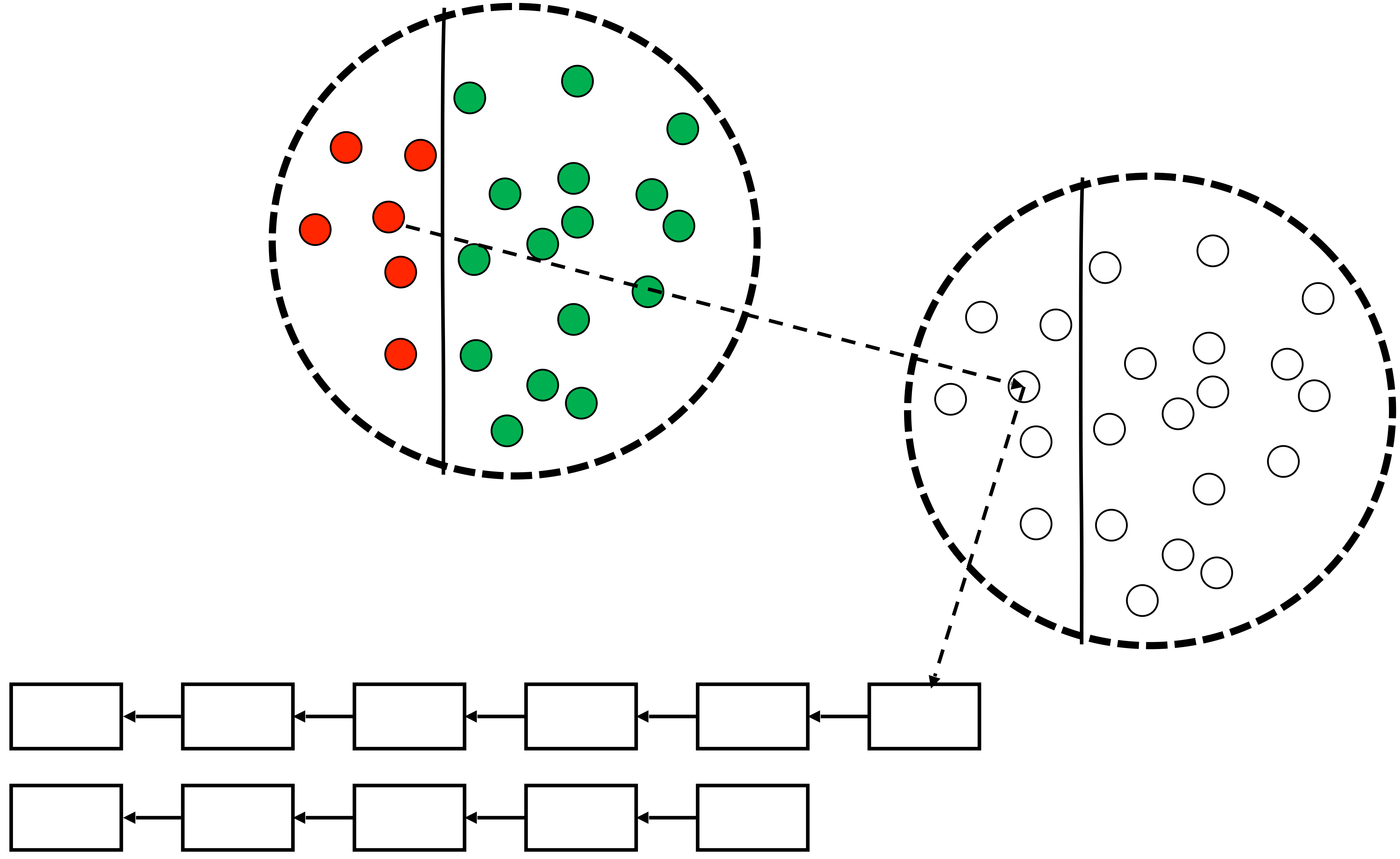
PoW/PoS-based Blockchain



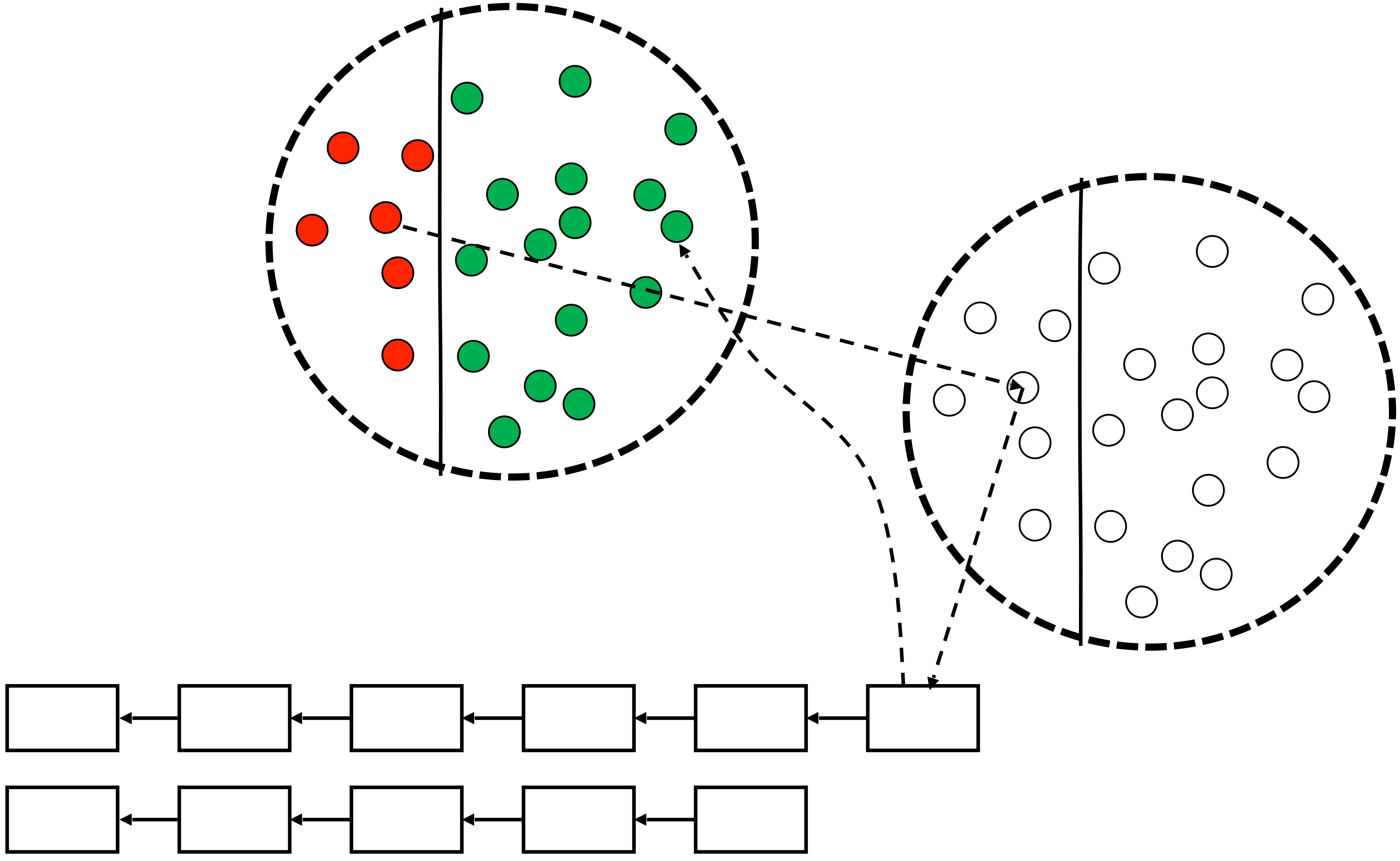
PoW/PoS-based Blockchain



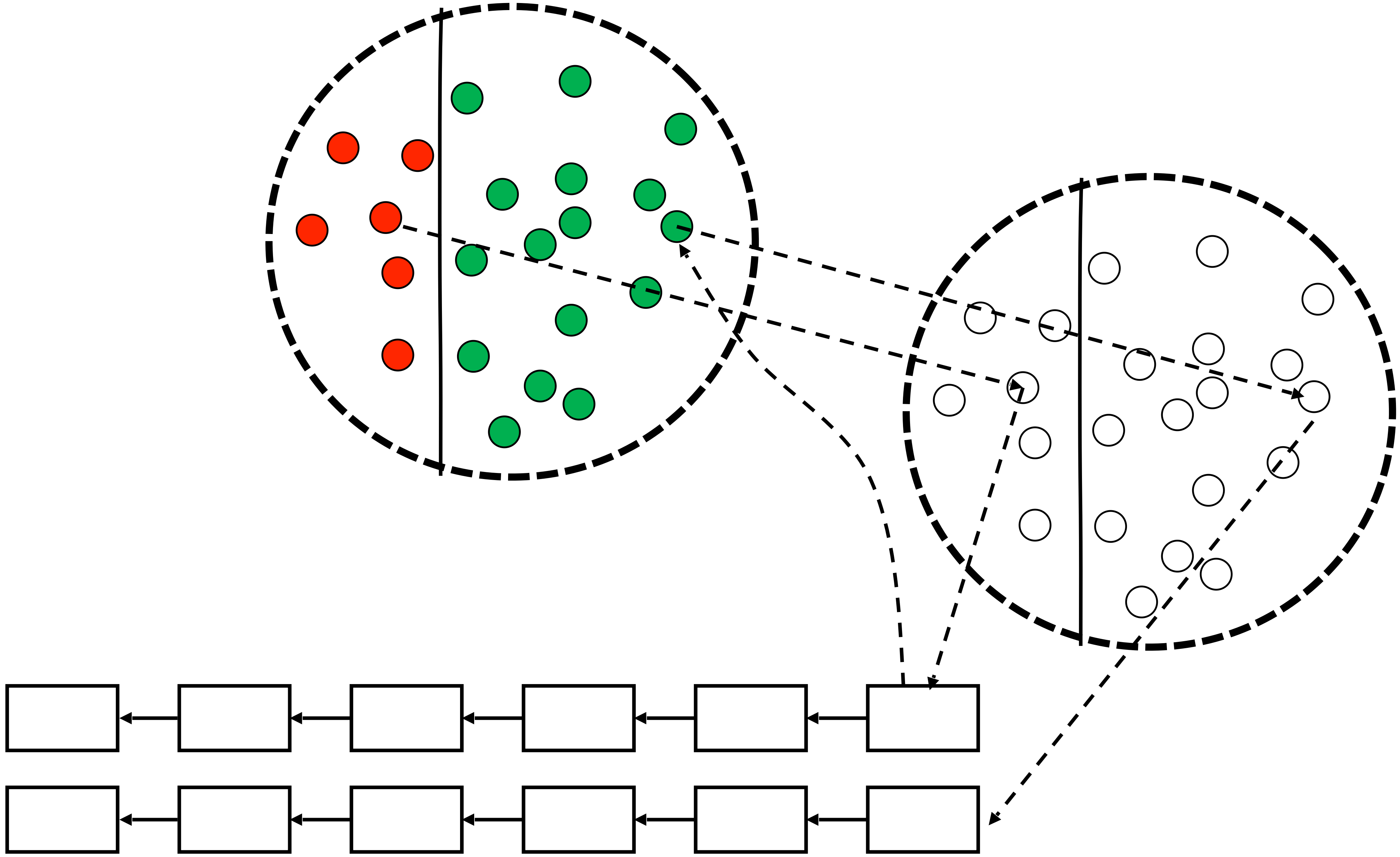
PoW/PoS-based Blockchain



PoW/PoS-based Blockchain

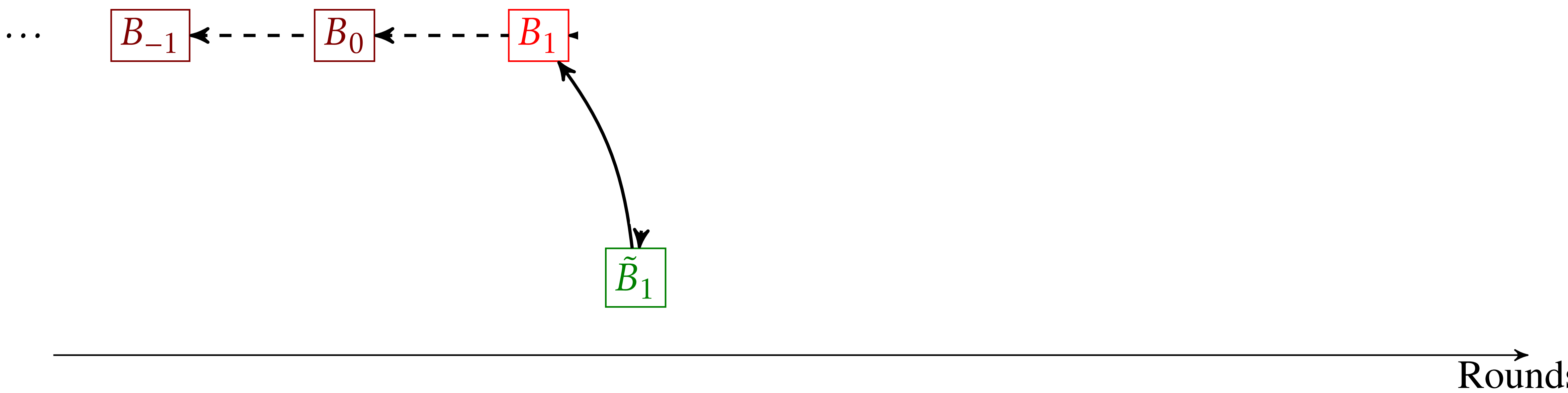


PoW/PoS-based Blockchain



2-hop blockchain

[Duong,Fan,**Z.**,16]

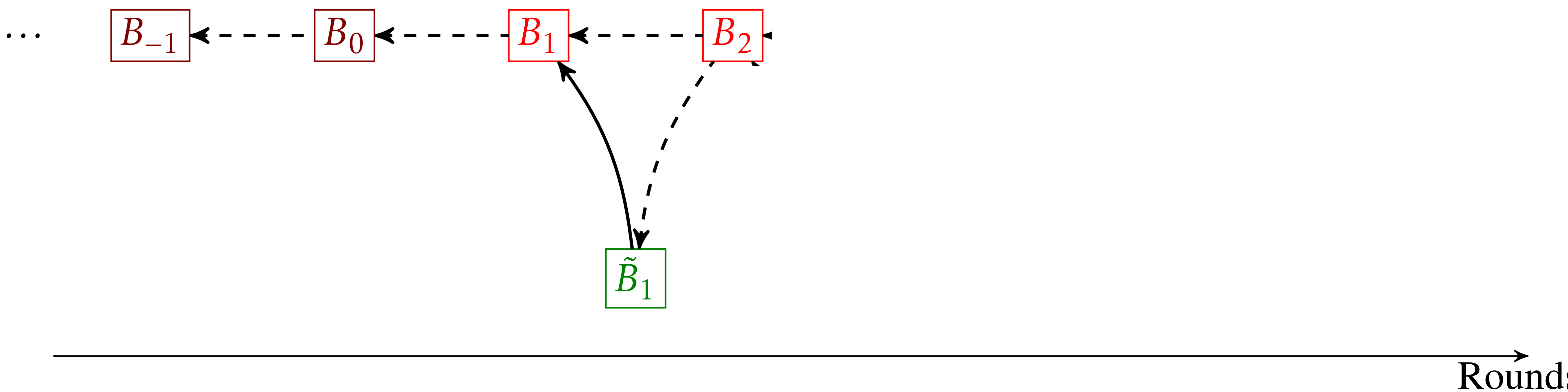


$$H(B_1||\tilde{B}_1||nonce_1) < \mathtt{T}$$

Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

2-hop blockchain

[Duong,Fan,**Z**.,16]

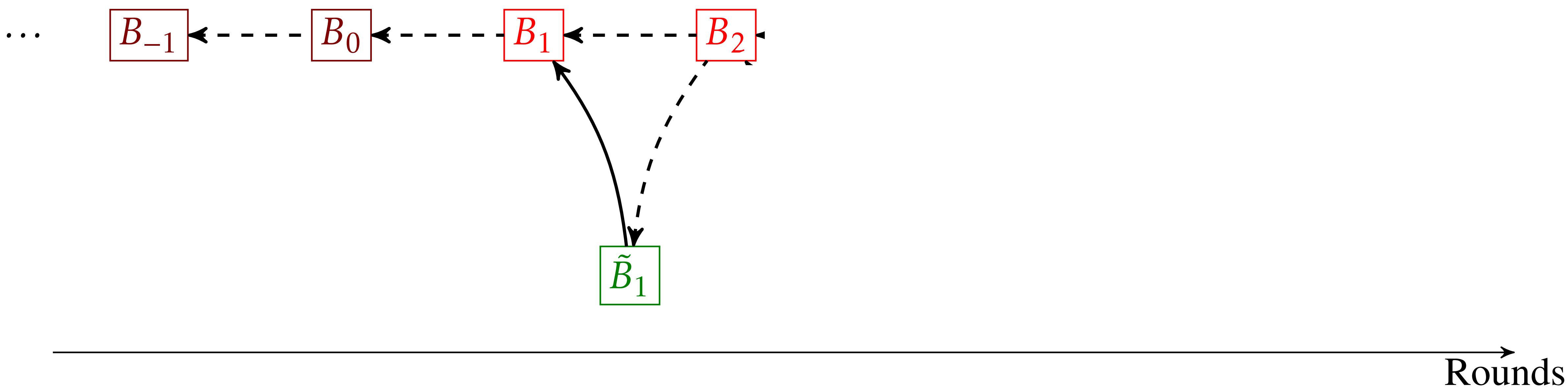


$$H(B_1||\tilde{B}_1||nonce_1) < \mathtt{T}$$

Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

2-hop blockchain

[Duong,Fan,**Z**.,16]



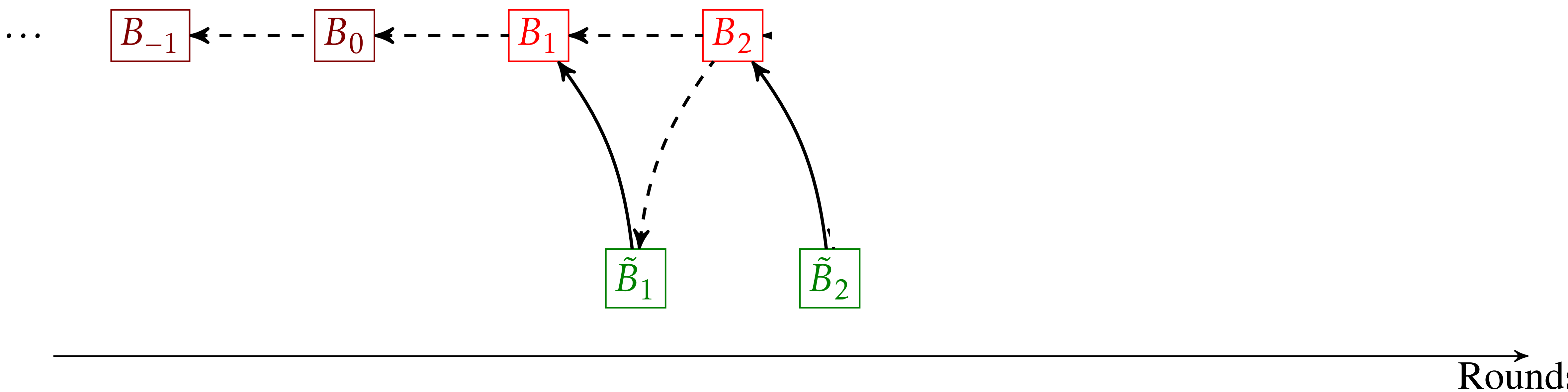
$$H(B_1 || \tilde{B}_1 || nonce_1) < \mathtt{T}$$

$$\tilde{H}(B_2 || \tilde{\mathsf{vk}}_2) < \tilde{\mathtt{T}}$$

Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

2-hop blockchain

[Duong,Fan,**Z.**,16]

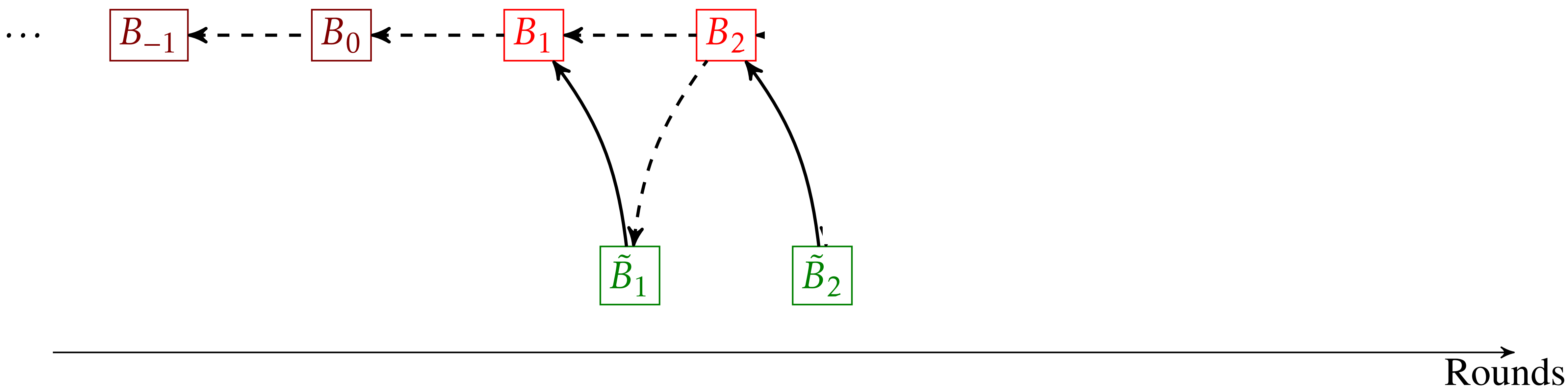


$$H(B_1||\tilde{B}_1||nonce_1) < \mathtt{T} \qquad \tilde{H}(B_2||\tilde{\mathsf{vk}}_2) < \tilde{\mathtt{T}}$$

Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

2-hop blockchain

[Duong,Fan,**Z**.,16]



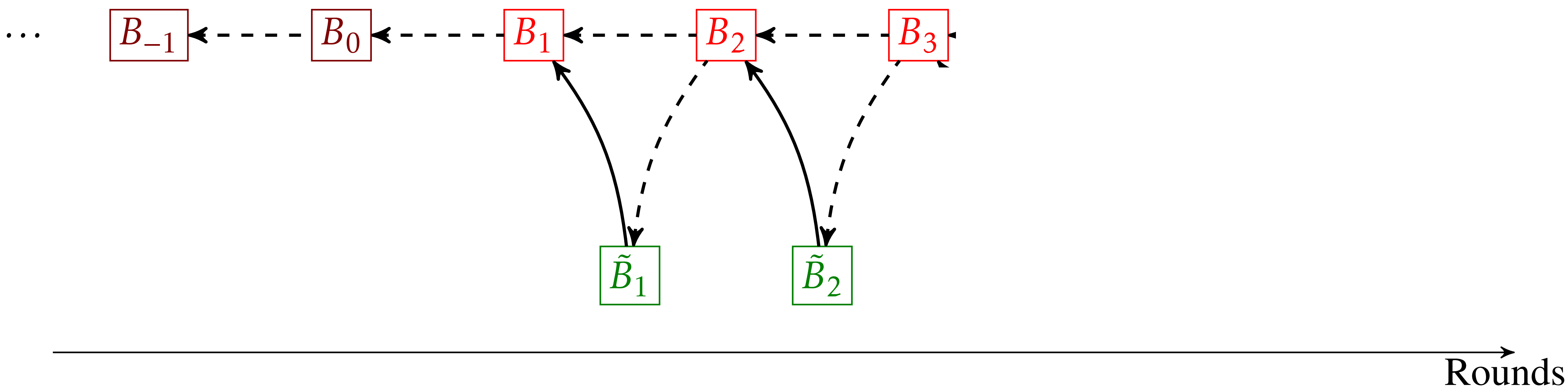
$$H(B_1||\tilde{B}_1||nonce_1) < \mathtt{T}$$
$$H(B_2||\tilde{B}_2||nonce_2) < \mathtt{T}$$

$$\tilde{H}(B_2||\tilde{\mathsf{vk}}_2) < \tilde{\mathtt{T}}$$

Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

2-hop blockchain

[Duong,Fan,**Z**.,16]



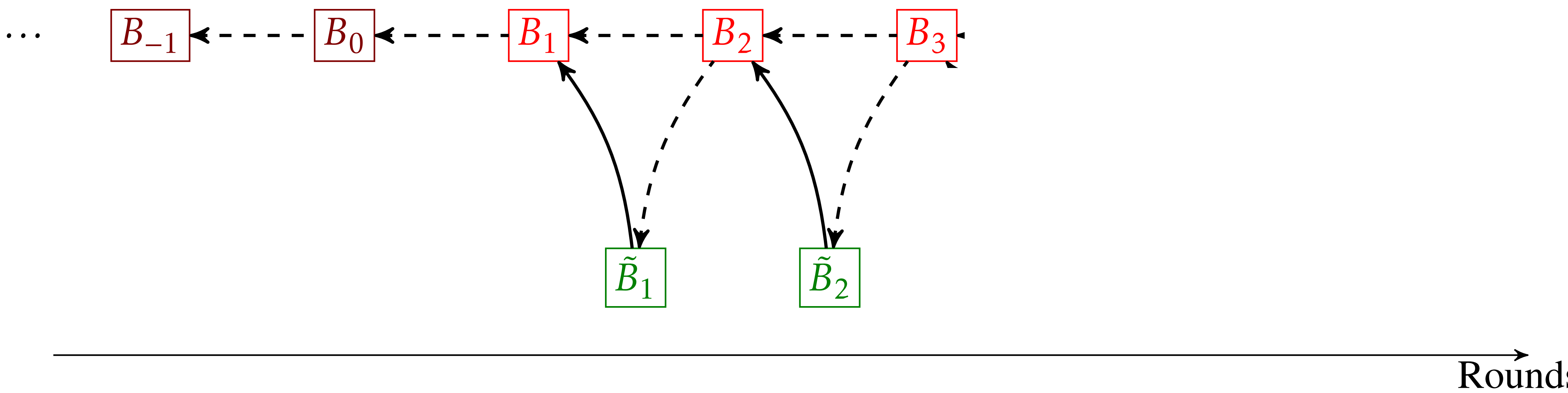
$$H(B_1||\tilde{B}_1||nonce_1) < \mathtt{T}$$
$$H(B_2||\tilde{B}_2||nonce_2) < \mathtt{T}$$

$$\tilde{H}(B_2||\tilde{\mathsf{vk}}_2) < \tilde{\mathtt{T}}$$

Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

2-hop blockchain

[Duong,Fan,**Z.**,16]



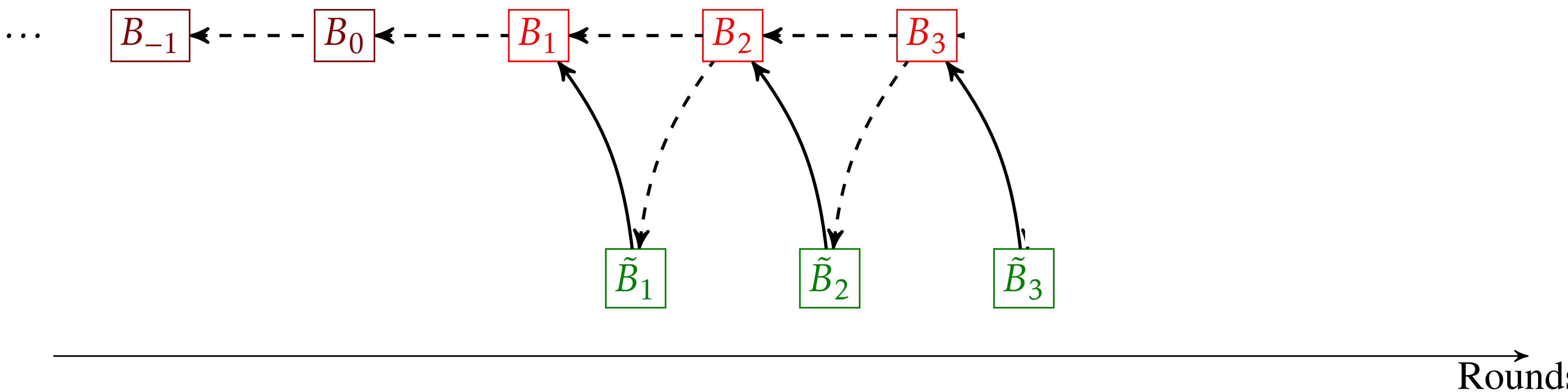
$$H(B_1||\tilde{B}_1||nonce_1) < \mathtt{T} \qquad \tilde{H}(B_2||\tilde{\mathsf{vk}}_2) < \tilde{\mathtt{T}}$$

$$H(B_2||\tilde{B}_2||nonce_2) < \mathtt{T} \qquad \tilde{H}(B_3||\tilde{\mathsf{vk}}_3) < \tilde{\mathtt{T}}$$

Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

2-hop blockchain

[Duong,Fan,**Z.**,16]



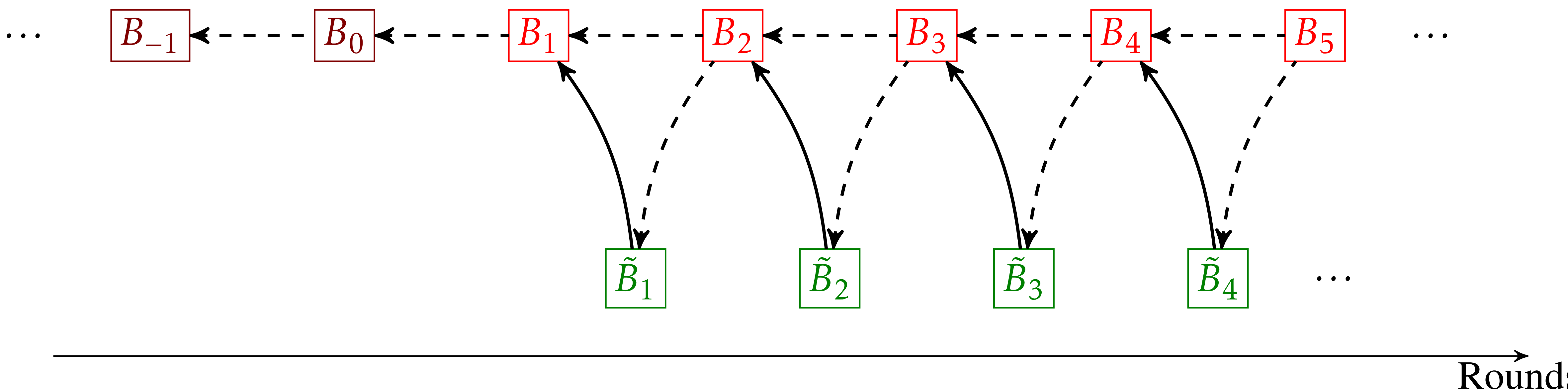
$$H(B_1||\tilde{B}_1||nonce_1) < \mathtt{T}$$
$$H(B_2||\tilde{B}_2||nonce_2) < \mathtt{T}$$

$$\tilde{H}(B_2||\tilde{\mathsf{vk}}_2) < \tilde{\mathtt{T}}$$
$$\tilde{H}(B_3||\tilde{\mathsf{vk}}_3) < \tilde{\mathtt{T}}$$

Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

2-hop blockchain

[Duong,Fan,**Z.**,16]

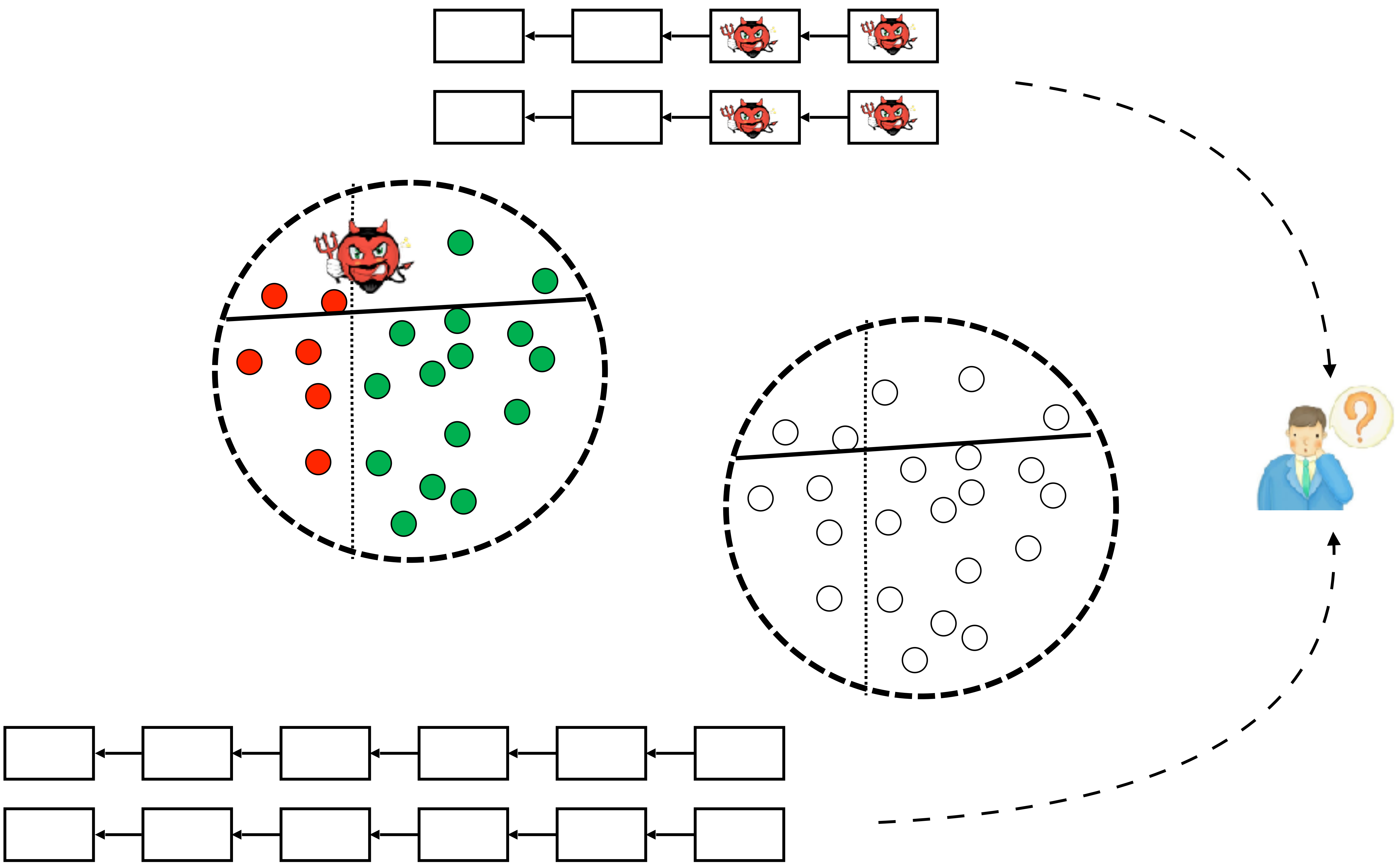


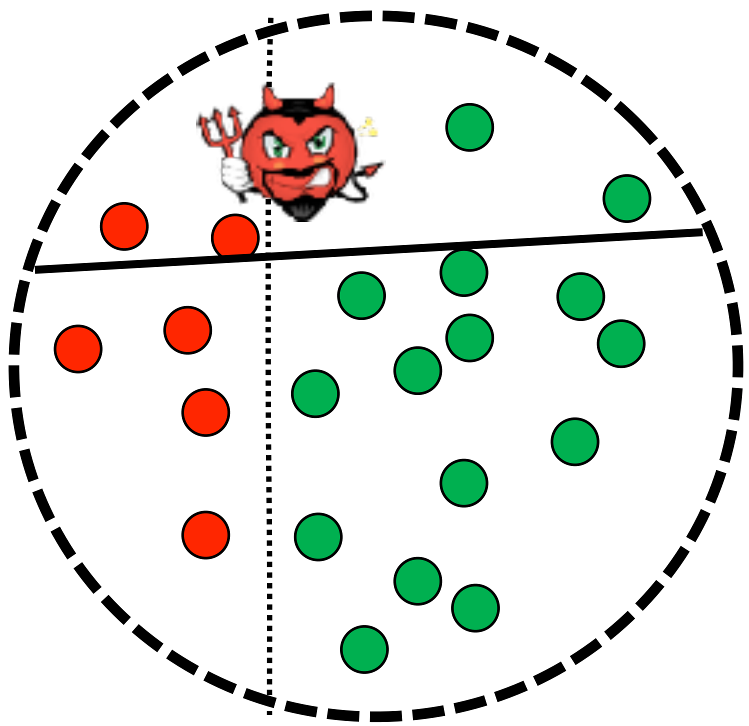
$$H(B_1||\tilde{B}_1||nonce_1) < \mathtt{T} \qquad \tilde{H}(B_2||\tilde{\mathsf{vk}}_2) < \tilde{\mathtt{T}}$$

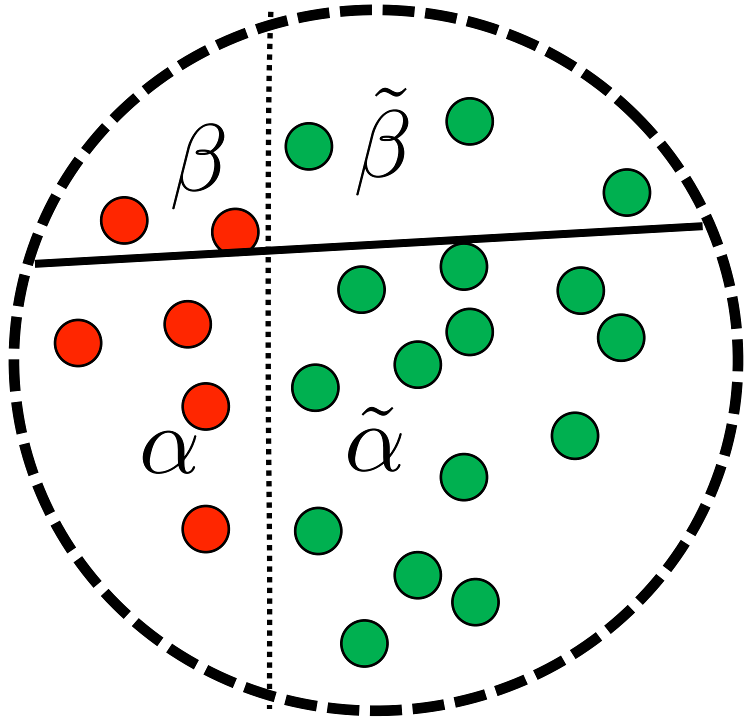
$$H(B_2||\tilde{B}_2||nonce_2) < \mathtt{T} \qquad \tilde{H}(B_3||\tilde{\mathsf{vk}}_3) < \tilde{\mathtt{T}}$$

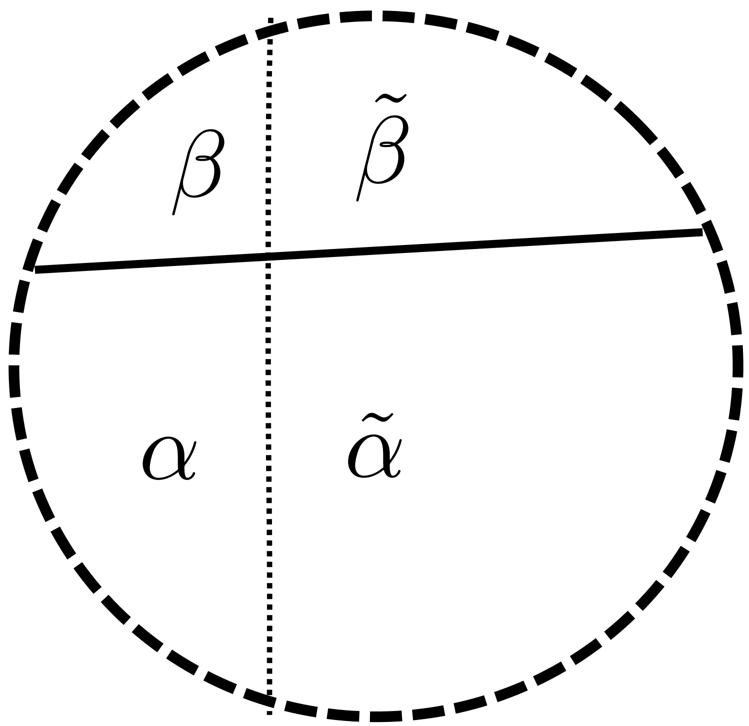
Duong, Fan, Zhou, 2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely. IACR ePrint 2016

PoW/PoS-based Blockchain






β
$\tilde{\beta}$
α
$\tilde{\alpha}$



$$\hat{\beta} = \beta\tilde{\beta}$$

$$\hat{\alpha} = \alpha\tilde{\alpha}$$

$$\hat{\beta} < \hat{\alpha}$$

2-hop Blockchain

[Duong,Fan,**Z**.,16]

- Scalable to a huge network of nodes
- provably secure

V.0

2-hop Blockchain

[Duong,Fan,**Z**.,16]

- Scalable to a huge network of nodes

- provably secure

51% Honest Mining Power Assumption could be challenged

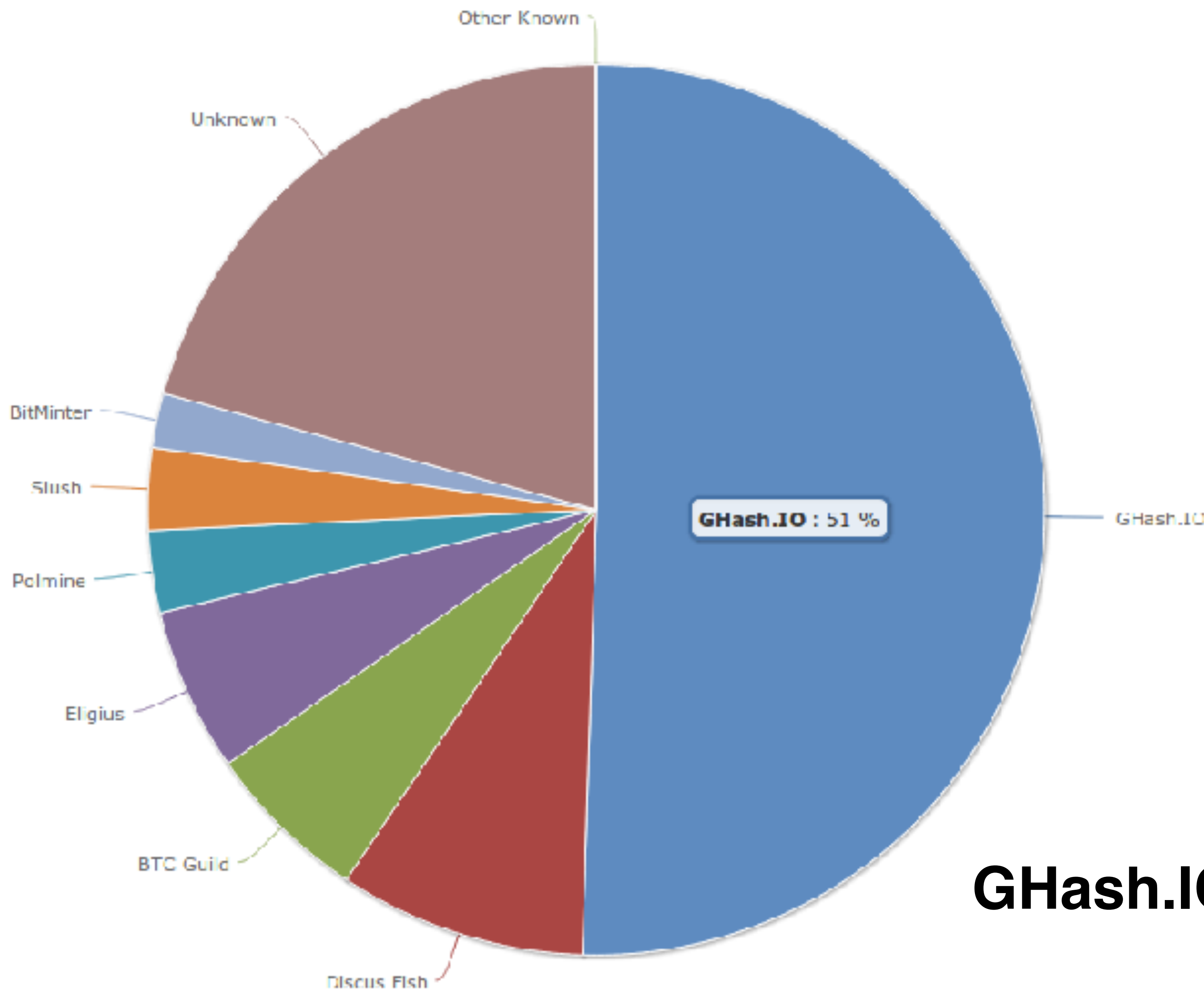
What about dedicated hardware?

黑科技

each new technique may have two sides

a breakthrough in certain area may be a big disaster in other areas

51% Honest Mining Power Assumption could be challenged



June 12, 2014
GHash.IO large mining pool crisis

All top mining pools are in China!

They might collude for whatever reason

V.1

TwinsChain

[Chepurnoy,Duong,Fan,**Z**.,16]

- Scalable to a huge network of nodes
- provably secure
- adaptive difficulty adjustment
- implementation

Chepurnoy, Duong, Fan, Zhou, TwinsCoin: A Cryptocurrency via Proof-of-Work and Proof-of-Stake. IACR ePrint 2017

- adaptive difficulty adjustment

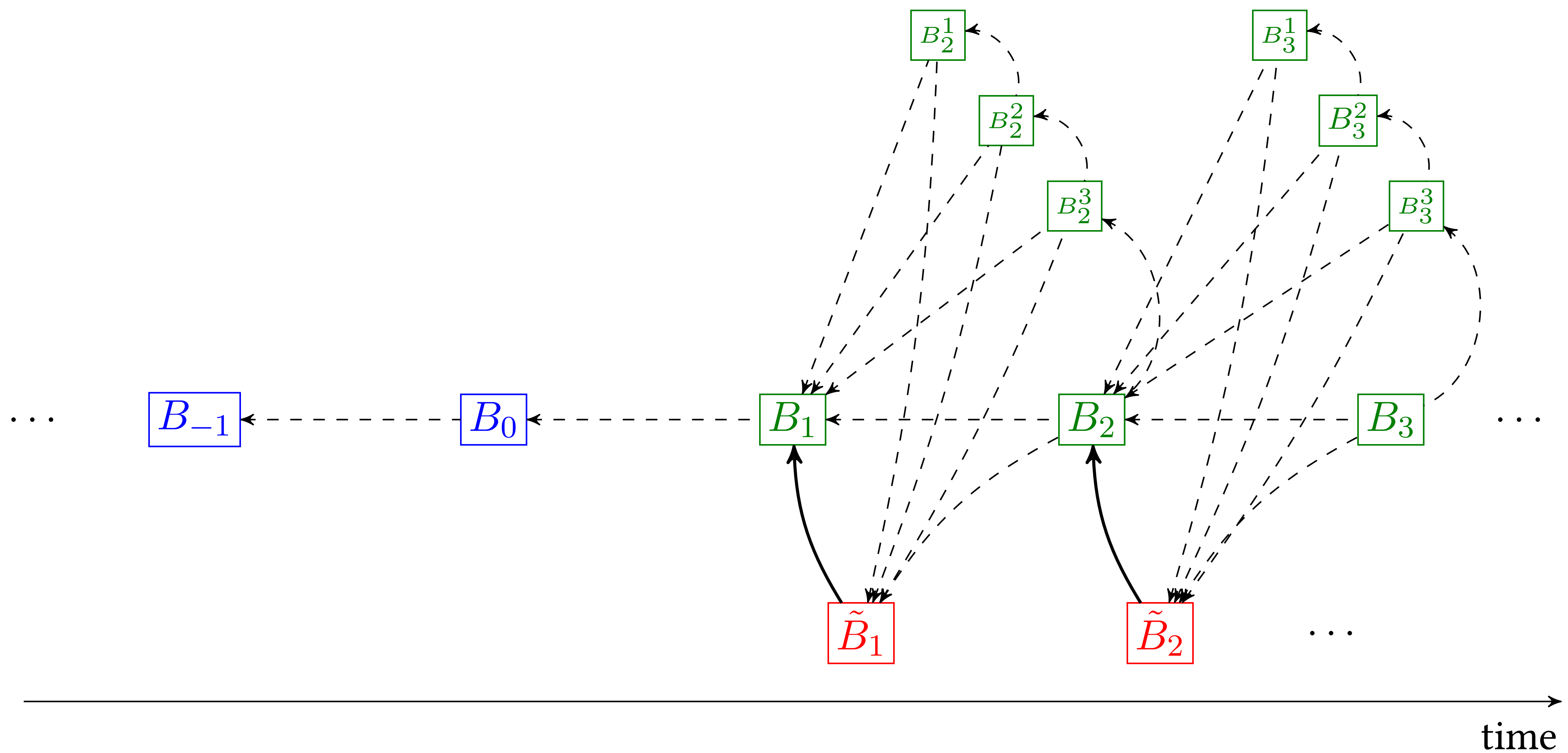


Figure 2: TwinsChain blockchain structure

Here, dot arrows (that link to the previous successful block and attempting blocks) denote the first hops, and solid arrows denote the second hops. Green blocks B_i's denote the successful proof-of-work blocks, B_i^j's denote the attempting proof-of-work blocks, and red blocks $\tilde{B}_i$'s denote the corresponding proof-of-stake blocks. Note that the blue blocks are from the "mature blockchain".

Atlassian, Inc.

Inbox (2,268) - hszhou@vcu.edu - Virginia C... | Bitcoin Summer School 2016 | Cryptology ePrint Archive: Recent Updates | twinscoin / twinschain

Bitbucket Teams Projects Repositories Snippets

twinschain

ACTIONS

- Clone
- Compare
- Fork

NAVIGATION

- Overview
- Source
- Commits
- Branches
- Pull requests
- Downloads

twinschain / twinschain

Overview

https://bitbucket.org/twinscoin/twinschain

Last updated	2016-11-11
Language	—
Access level	Read

1 Branch | 0 Tags | 0 Forks | 1 Watcher

TwinsChain Implementation

Building

The following instruction are for Linux environment only.

1. First, you need to install Java. Oracle Java 8 is preferred. For Ubuntu, you can follow this article http://tecadmin.net/install-oracle-java-8-jdk-8-ubuntu-via-ppa/, for other distribution, download from the Oracle website and install manually.
2. Scala Build Tool (SBT) is needed. Follow download section at the website to install http://www.scala-sbt.org/download.html
3. We use snapshot versions for Scorex and IODB. Please build and publish them locally because of that

IODB:

```
git clone https://github.com/input-output-hk/iodb.git
sbt publishLocal
```

Scorex:

```
git clone https://github.com/ScorexFoundation/Scorex.git
sbt publishLocal
```

1. Build and run TwinsChain

how much is needed to mess up our system?

The simulation results show that even with 70% of total mining power an adversary also needs for about 20% of total stake to generate abetter chain than honest party's.

Given Bitcoin capitalization of ~ $80 billion, 20% of stake is about $16 billion.

V.2

TwinsChain

[Chepurnoy,Duong,Fan,**Z**.,16]

- Scalable to a huge network of nodes
- provably secure
- adaptive difficulty adjustment
- implementation
- incentives
- Mode switching
- Stress test

Alternative Mechanisms

A Design Technique:
Constructing blockchains via blockchains

i-hop Blockchain

- **i=1, other than Bitcoin**
- **i=2,**
- **i=3,**
- **….**

References

- BitcoinNG; Hybrid consensus; Elastico; ByzCoin;
- 2-hop blockchain;

Take home

- A unified view
- A design example
- A design technique

Cryptography on the Blockchain

the sky is the limit !

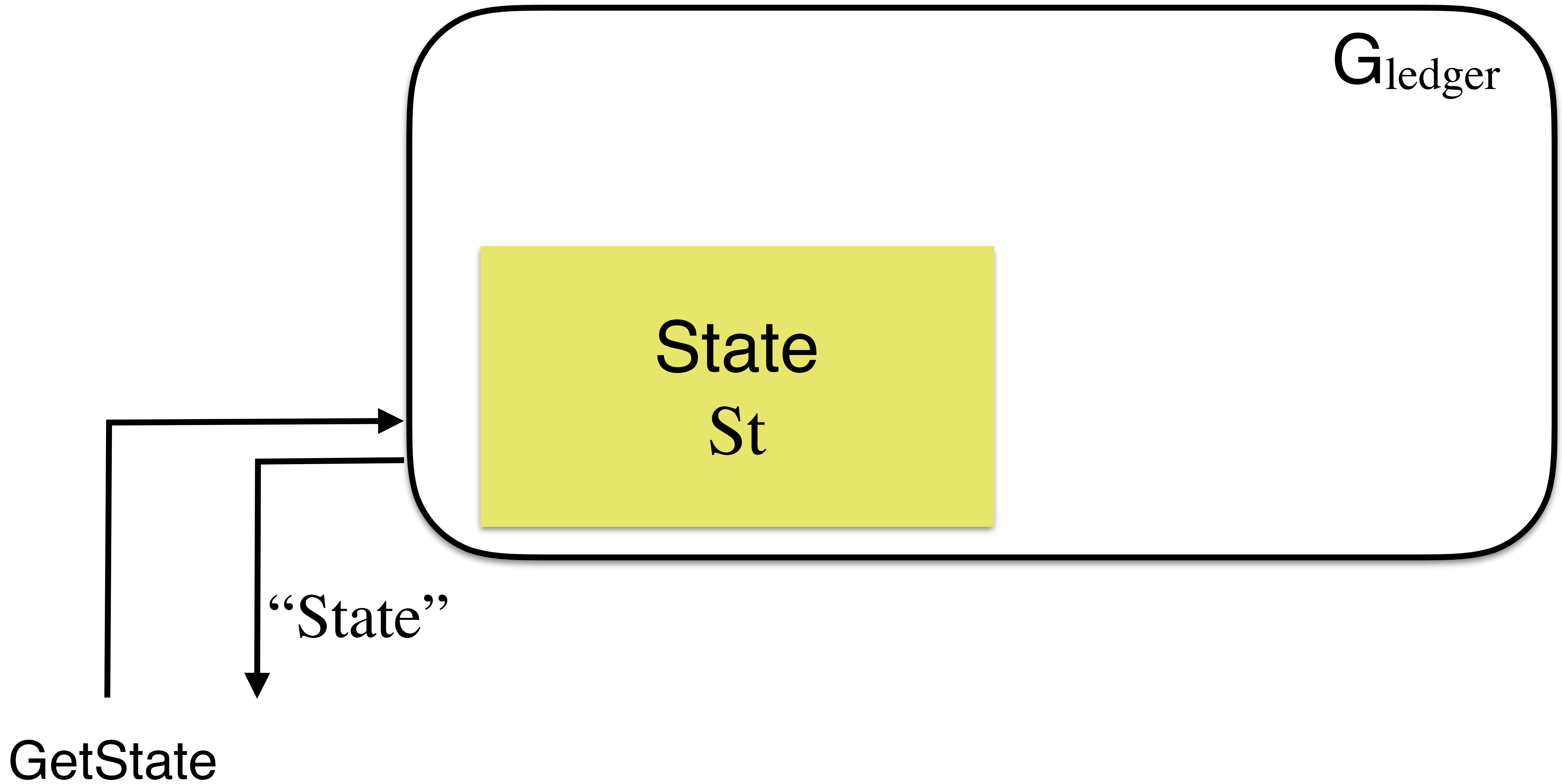
Fair Multi-Party Computation

- fairness = honest users get compensated by the adversarial users when the protocol aborts in an unfair manner

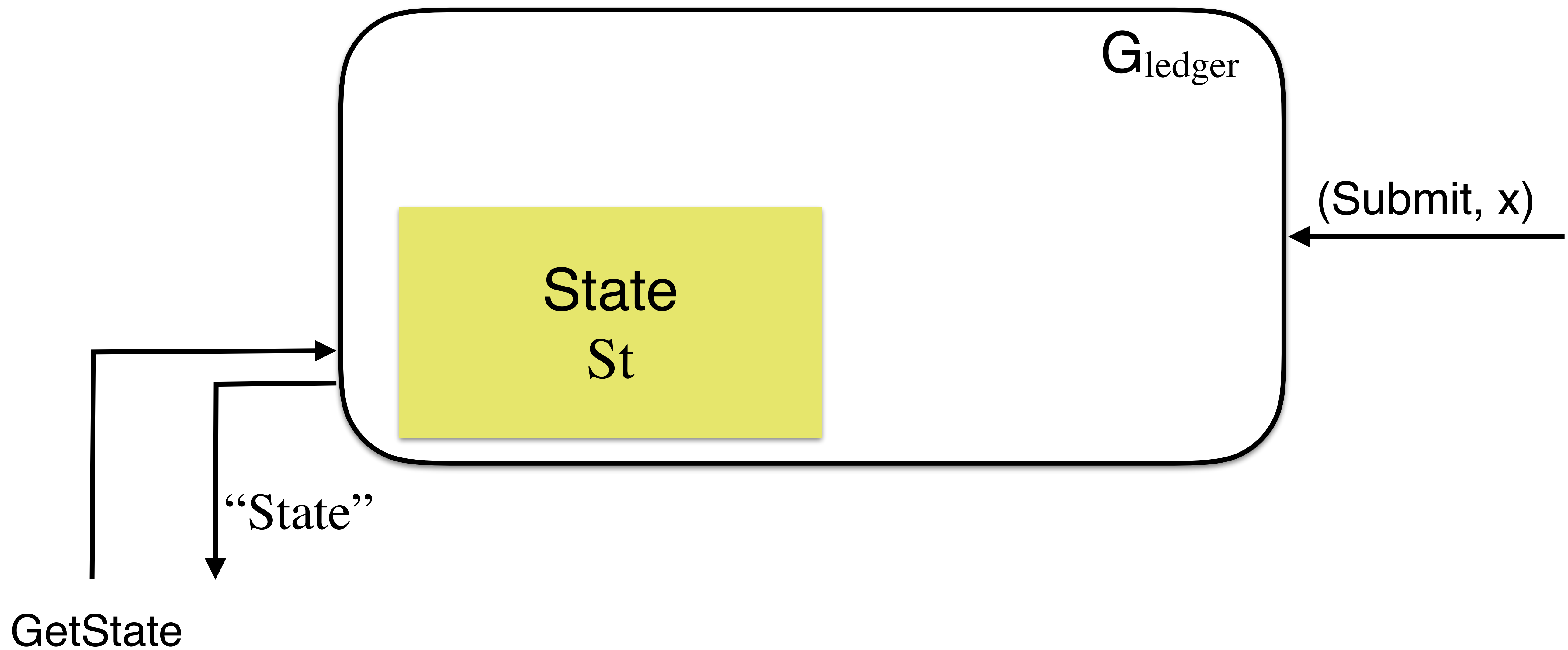
- first result that achieves such fairness for multi-party computation via blockchain with Universal Composability

Kiayias, Zhou, Zikas, *Fair and Robust Multi-Party Computation using a Global Transaction Ledger.* Eurocrypt 2016

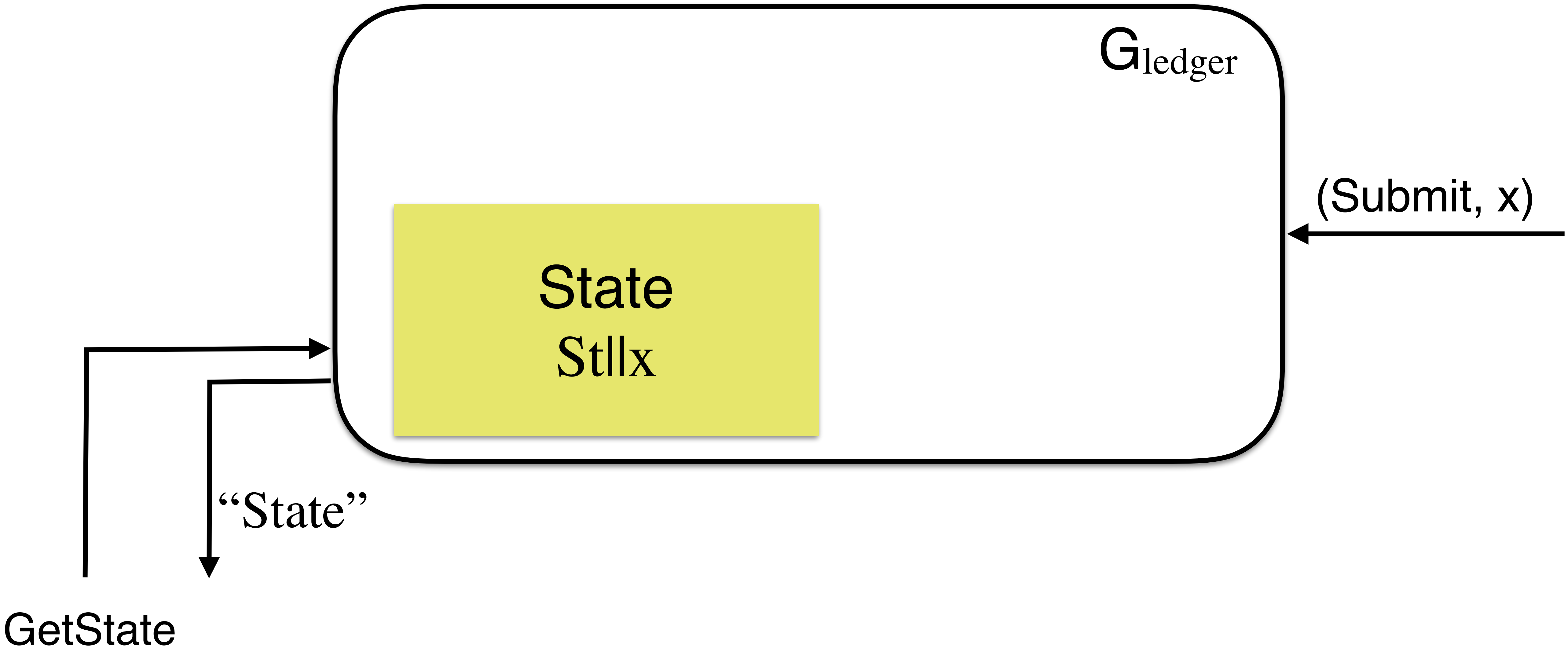
The Public Transaction Ledger



The Public Transaction Ledger

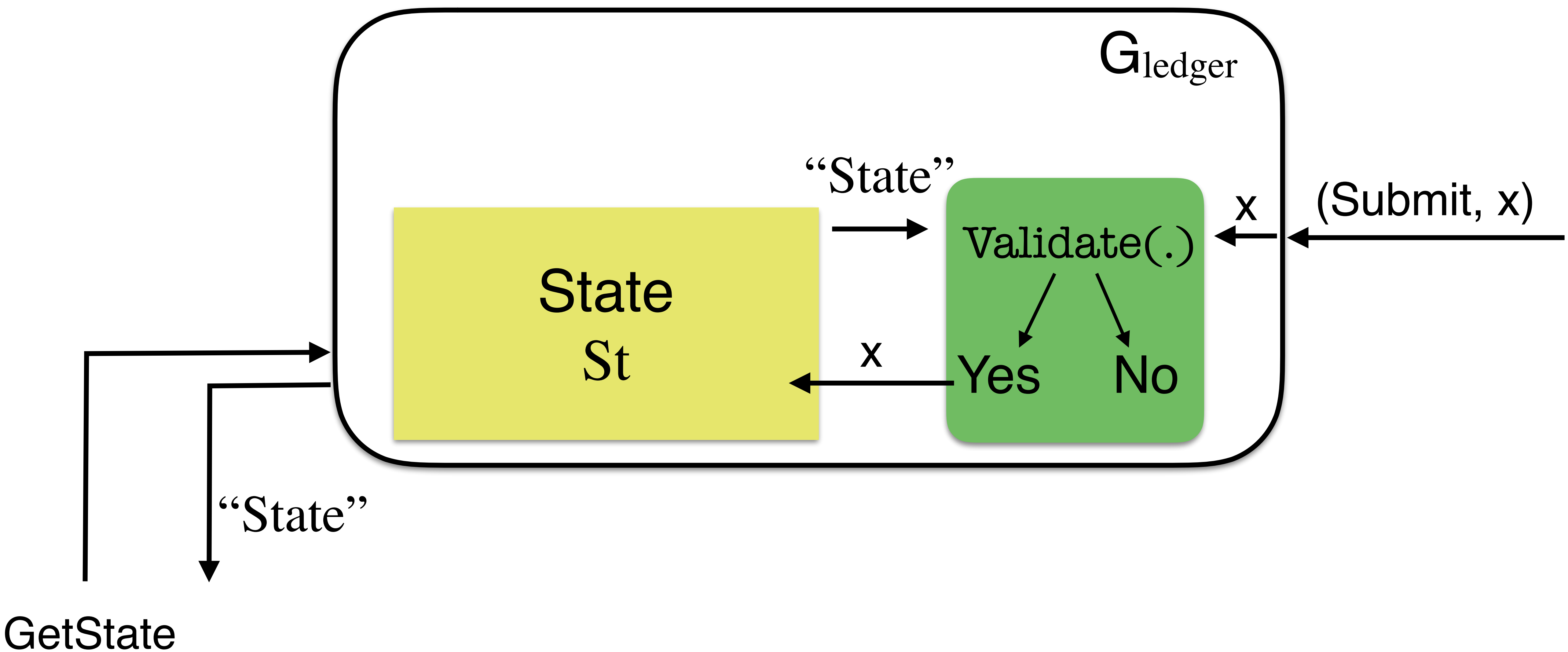


The Public Transaction Ledger



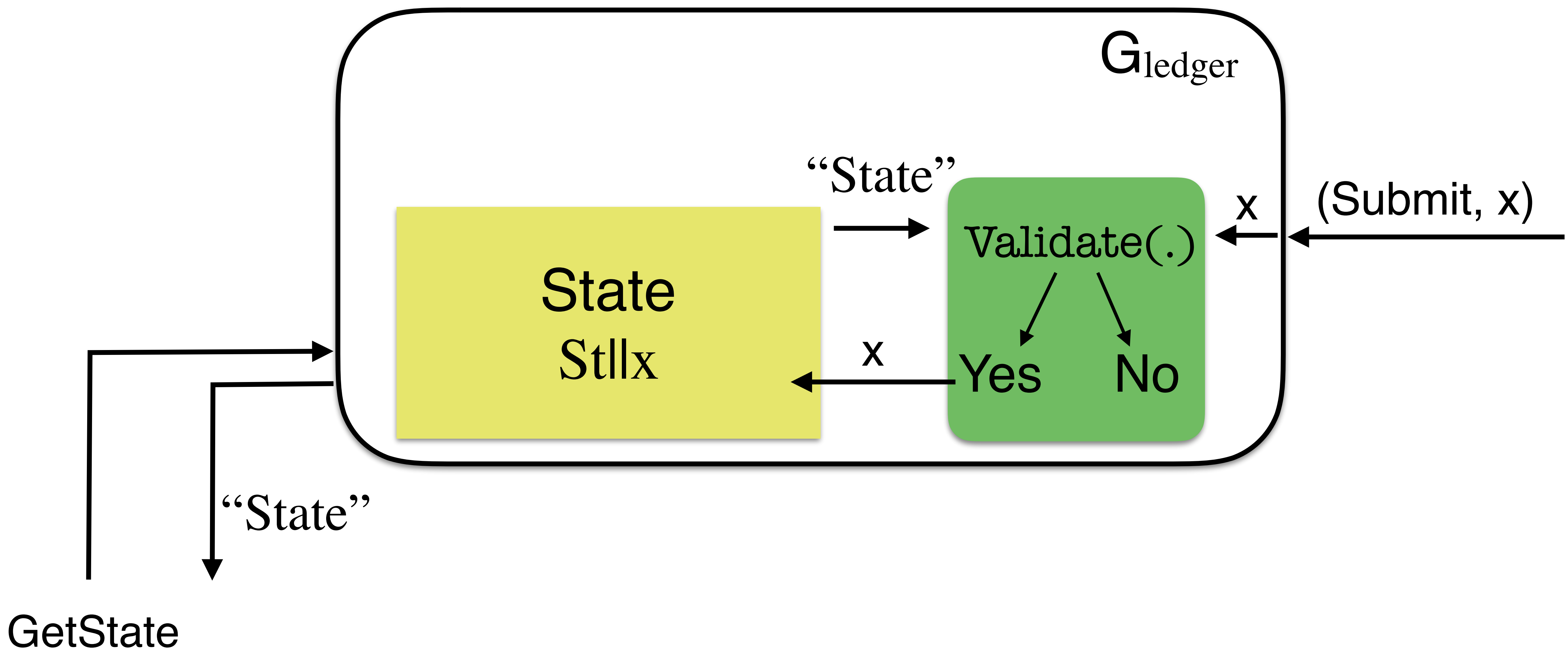
- In reality: Not a Bulletin Board
  - Inputs (transactions) are filtered

The Public Transaction Ledger



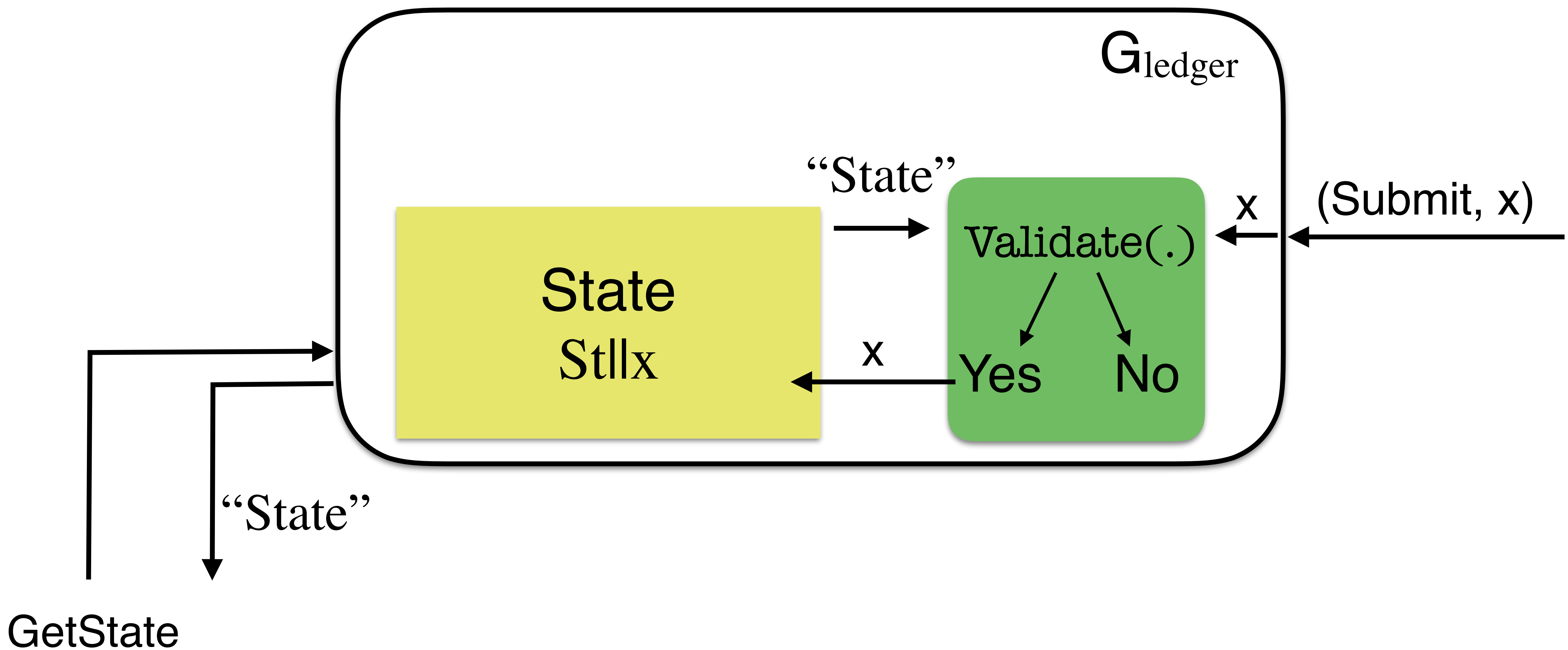
- In reality: Not a Bulletin Board
  - Inputs (transactions) are filtered

The Public Transaction Ledger



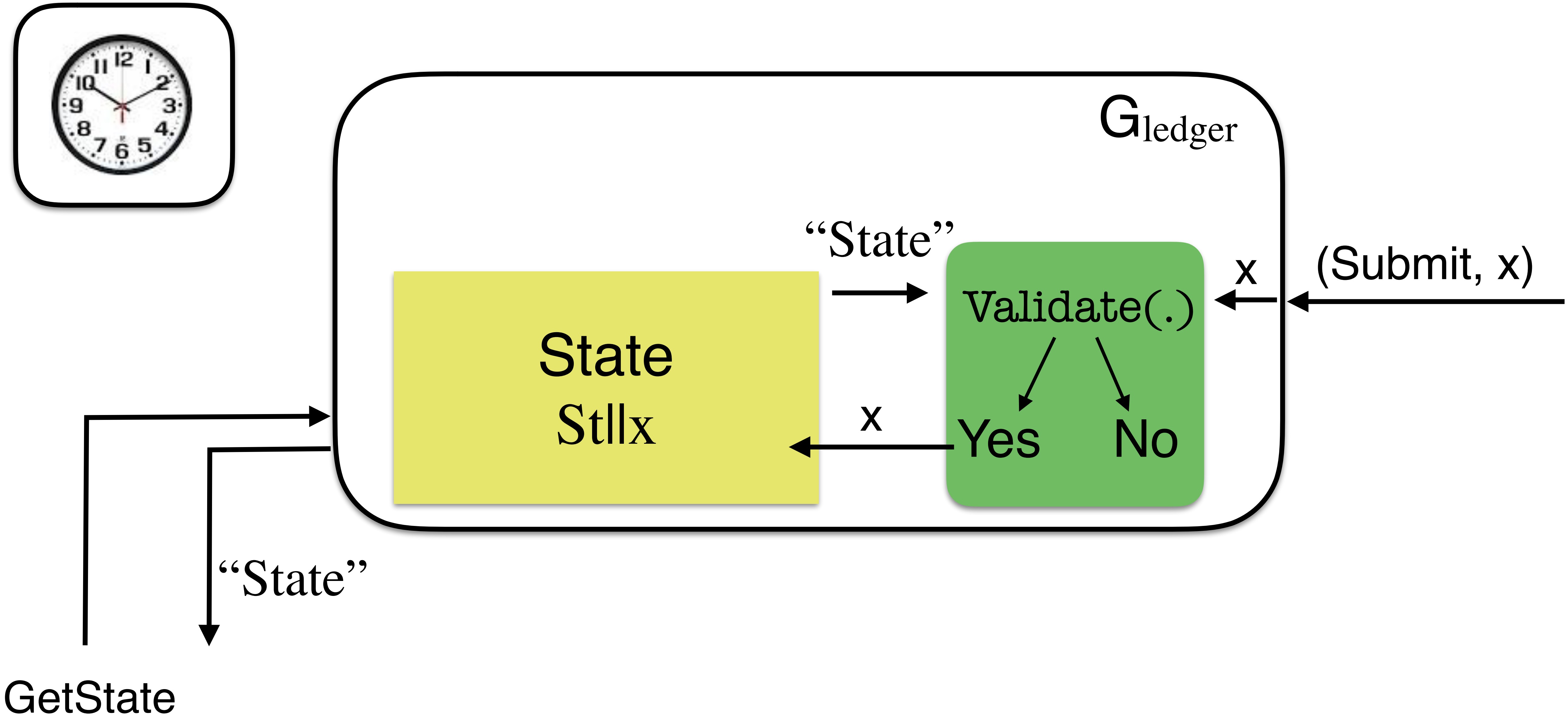
- In reality: Not a Bulletin Board
    - Inputs (transactions) are filtered

The Public Transaction Ledger



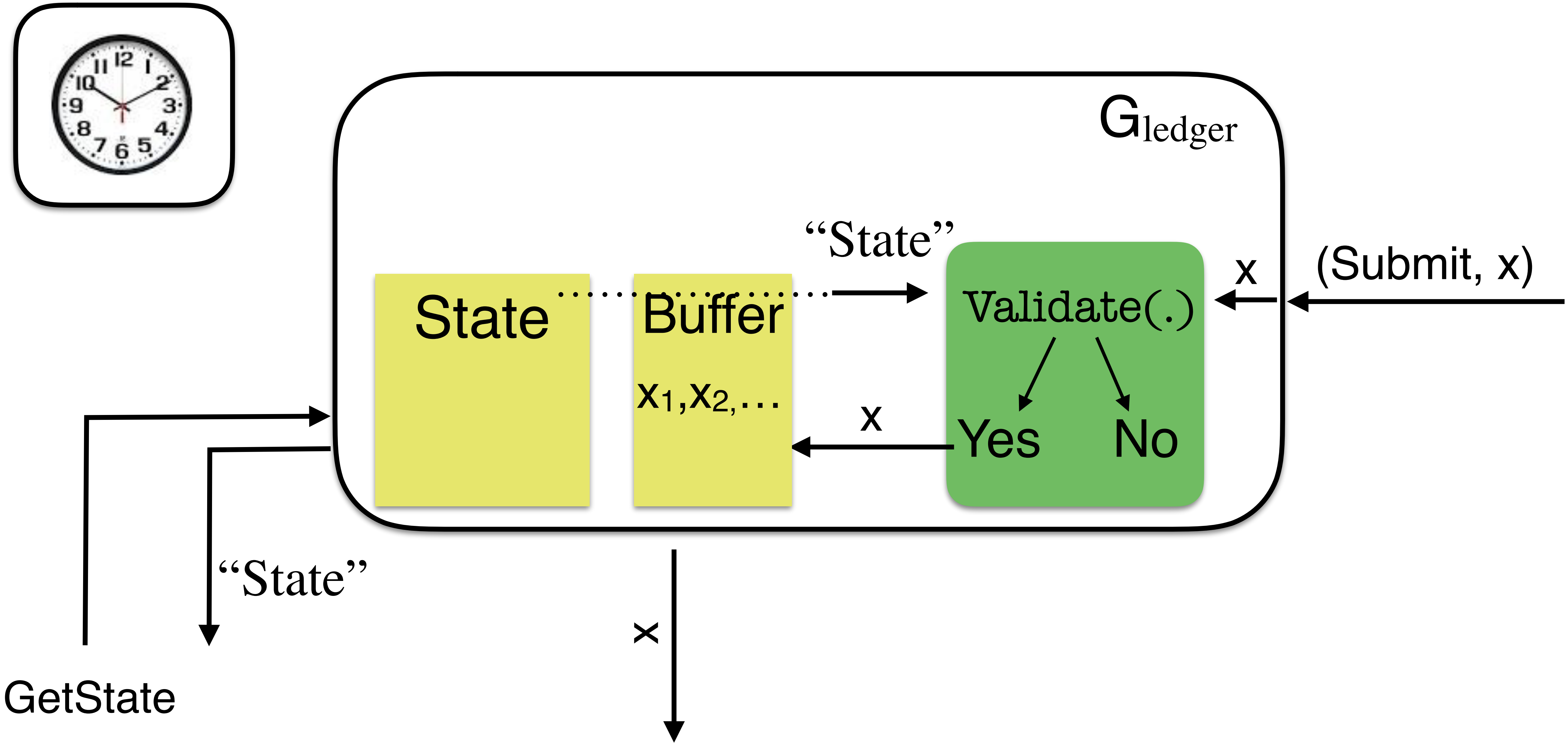
- In reality: Not a Bulletin Board
  - Inputs (transactions) are filtered
  - The order in which transactions in "State" are inserted might be adversarial … but not too adversarial

The Public Transaction Ledger



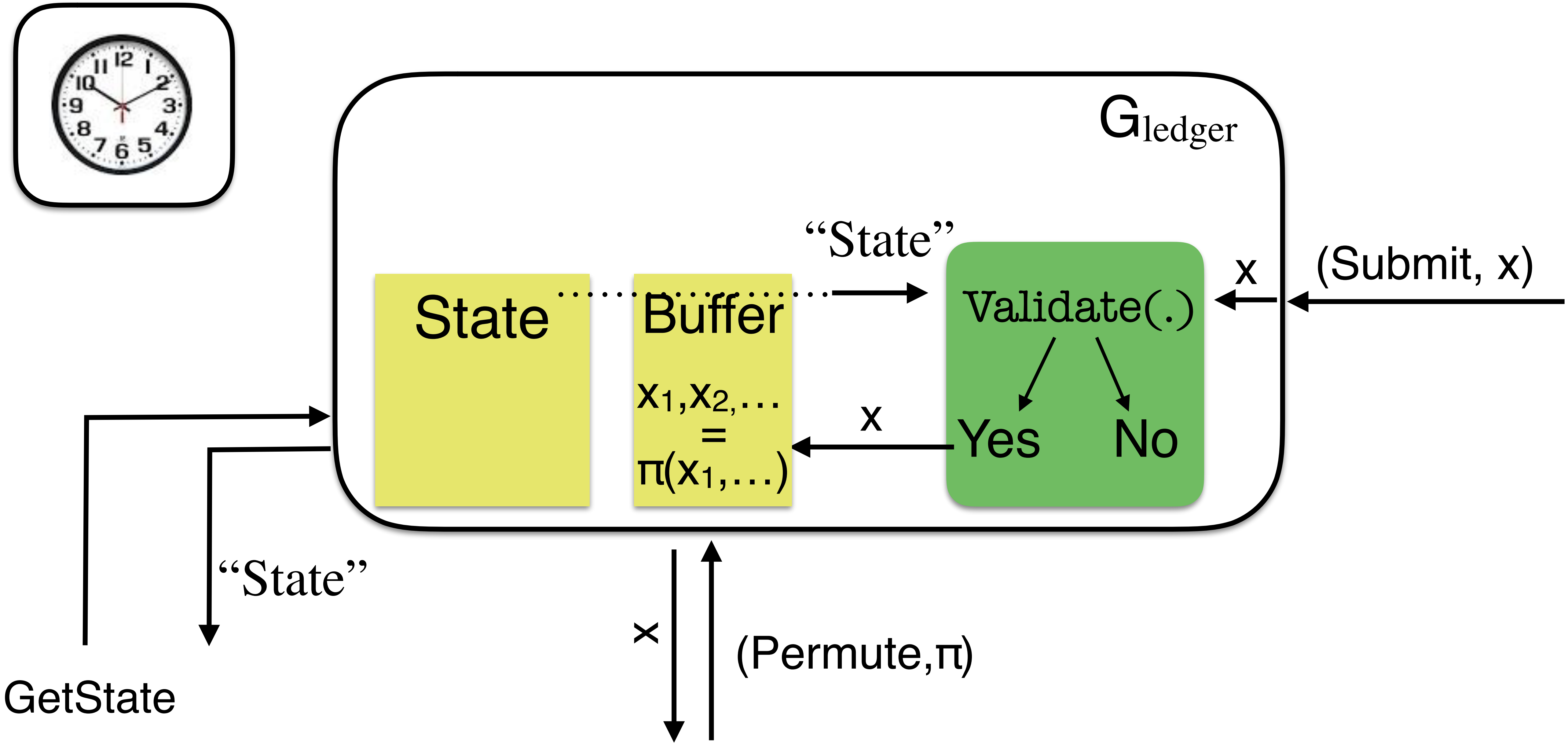
Can reorder **the recently inserted** transactions

The Public Transaction Ledger & Time



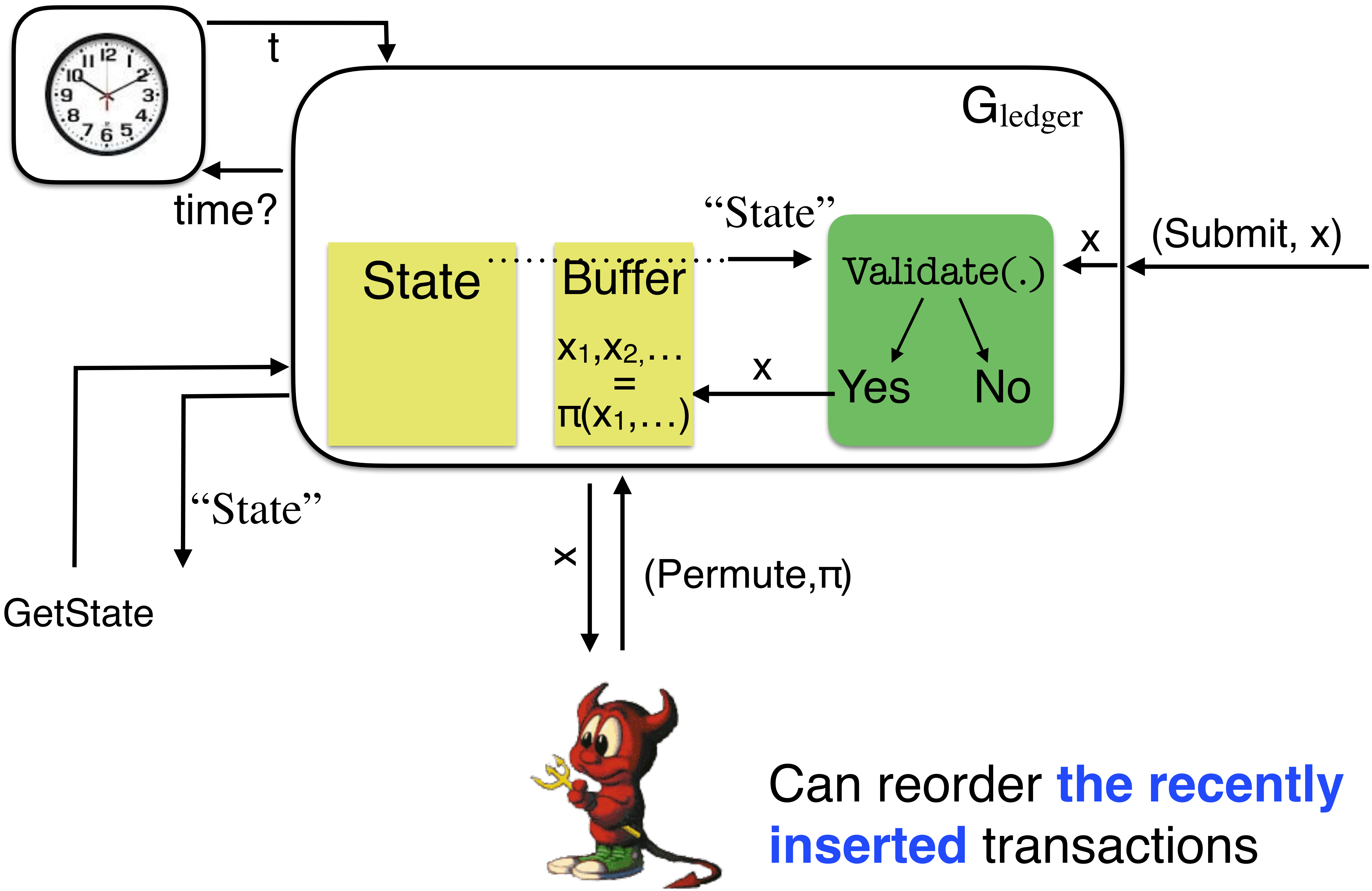
Can reorder **the recently inserted** transactions

The Public Transaction Ledger & Time



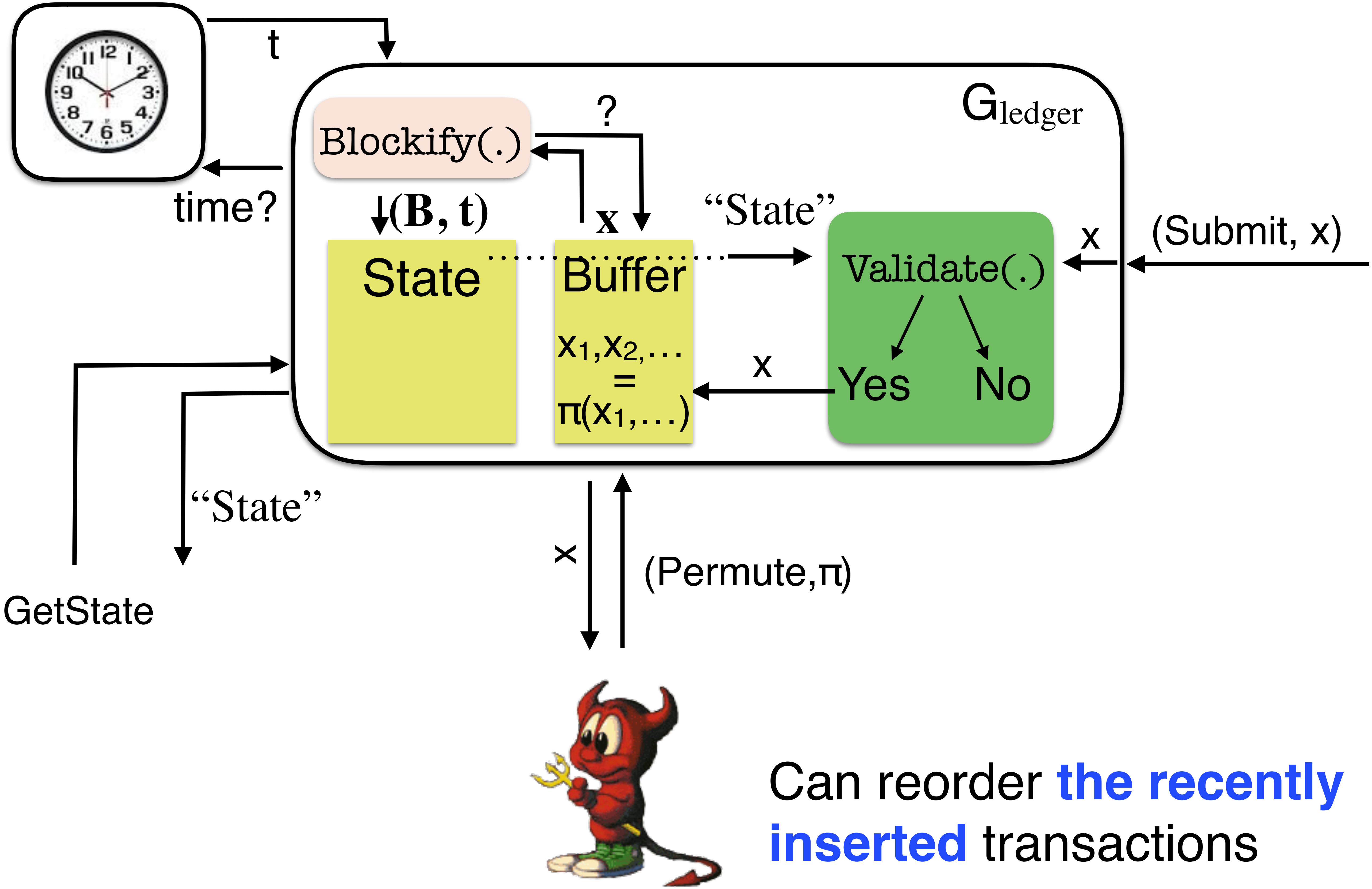
Can reorder **the recently inserted** transactions

The Public Transaction Ledger & Time



Can reorder **the recently inserted** transactions

The Public Transaction Ledger & Time



Can reorder **the recently inserted** transactions

my efforts along this line

- Zhang, Zhou, *Digital Liquid Democracy: How to Vote Your Delegation Statement.* IACR ePrint 2017. PODC 2017, brief announcement
- Kiayias, Zhou, Zikas, *Fair and Robust Multi-Party Computation using a Global Transaction Ledger.* Eurocrypt 2016
- Fan, Zhou, *iChing: A scalable proof-of-stake blockchain in the open setting (or, How to Mimic Nakamoto's Design via Proof-of-Stake).* IACR ePrint 2017
- Duong, Fan, Zhou, *2-hop Blockchain: Combining Proof-of-Work and Proof-of-Stake Securely*. IACR ePrint 2016
- Chepurnoy, Duong, Fan, Zhou, *TwinsCoin: A Cryptocurrency via Proof-of-Work and Proof-of-Stake*. IACR ePrint 2017
- Blocki, Zhou, *Designing Proof of Human-work Puzzles for Cryptocurrency and Beyond*. TCC 2016
- Duong, Zhou, Chepurnoy, *Multi-Mode Cryptocurrency Systems*. Manuscript.
- Katz, Garay, Kumaresan, Zhou, *Adaptively Secure Broadcast, Revisited*. PODC, 2011

https://cryptographylab.bitbucket.io/blockchain.html

Thanks

Hong-Sheng Zhou
Virginia Commonwealth University
https://cryptographylab.bitbucket.io
hongsheng.zhou@gmail.com