Analysis and Design of Blockchains

Rafael Pass

Based on [P-Seeman-Shelat] and [P-Shi]

Traditional distributed systems:
The “Permissioned” Model

- Consistency
- Liveness

Traditional distributed systems:
The “Permissioned” Model

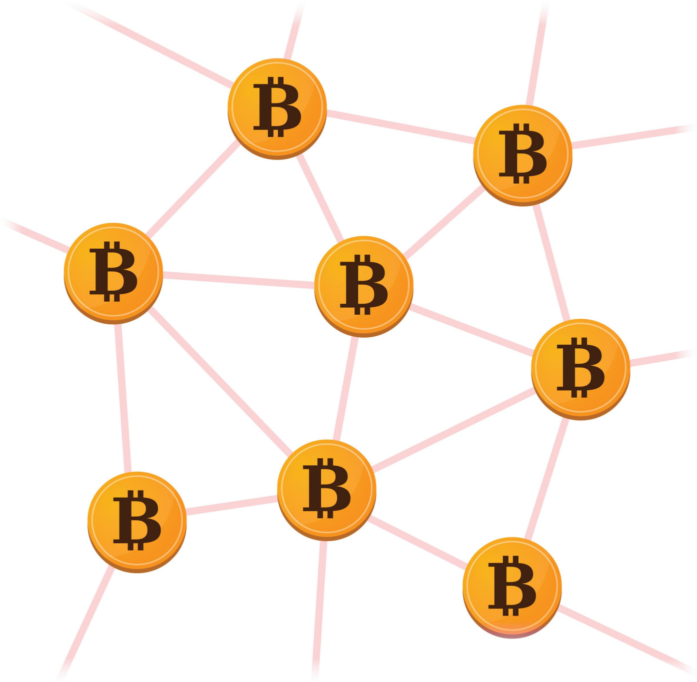
- Nodes a-priori known and authenticated

- 30 years of distributed systems

- Multi-party computation [GMW,BGW, ...]
  - Nearly all works assume authenticated channels

The "Permissionless" Model: Bitcoin/Blockchain

The Times 03/Jan/2009
Chancellor on brink of second bailout for banks.



The “Permissionless” Model

- Nodes do not know each other a-priori
- Nodes come and go
- ANYONE can join
- No network synchronization

Relatively little is known about this model

The “Permissionless” Model

- Strong impossibility results known in the “permissionless” (“unauthenticated”) model [BCLPR05]
  - **Consistency** is impossible
  - Sybil attacks unavoidable.
    - [BCLPR05] defined “weakened” security model (w/o consistency)

Nakamoto's Blockchain [Nak'08]

Prevents Sybil attacks with Proofs-of-Work Puzzles [DN'92]

Claims blockchain achieves "public ledger" assuming "honest majority":

- **Consistency**: everyone sees the same history
- **Liveness**: everyone can add new transactions

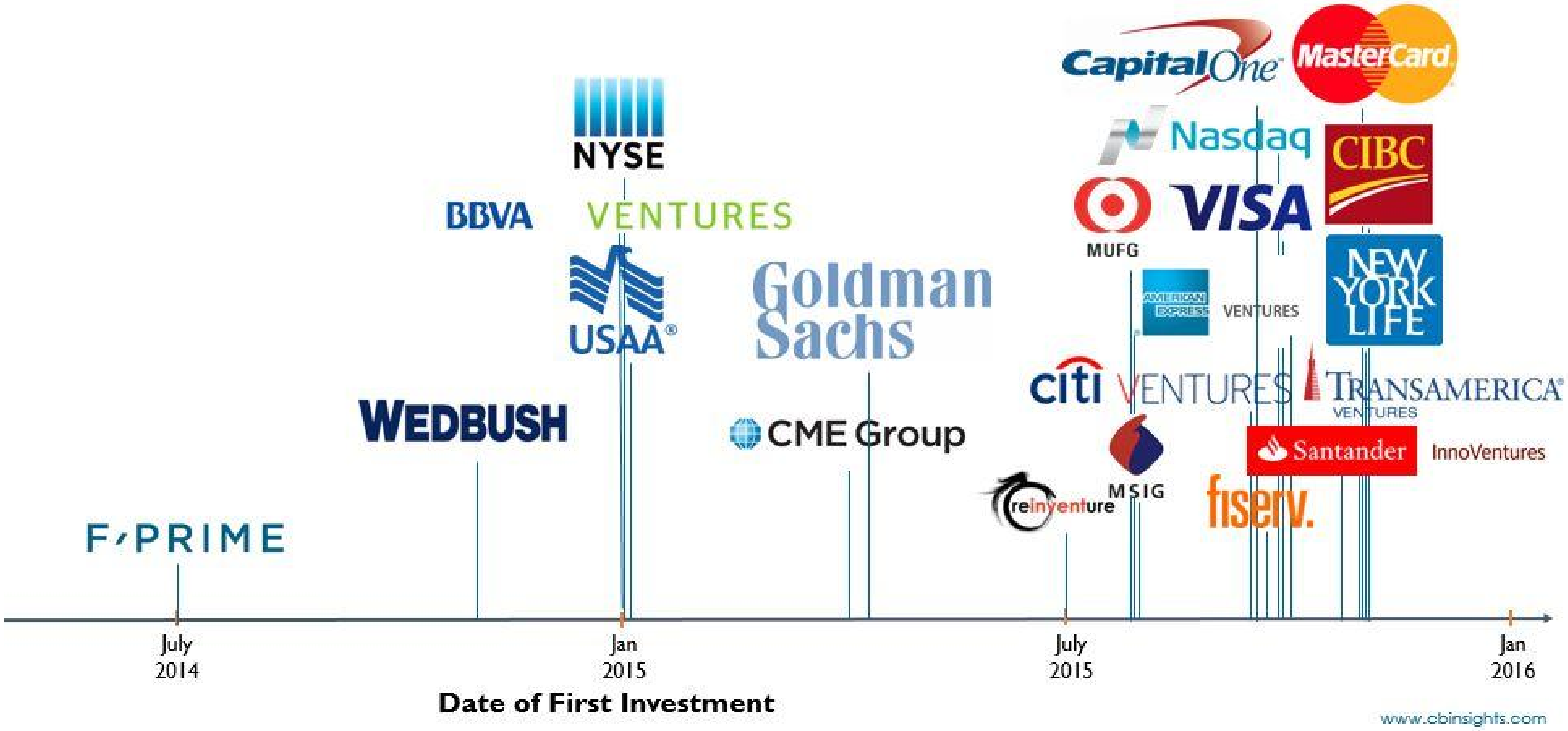
Nakamoto's Blockchain [Nak'08]

Prevents Sybil attacks with Proofs-of-Work Puzzles [DN'92]

2 amazing aspects:

- Overcomes permissionless barrier [BCLPR'05]
- Overcomes ⅓ barrier even in permissioned setting [LSP'83]

Everyone wants a “blockchain”



Nakamoto's Blockchain: OPEN PROBLEMS

- **WHAT IS** a blockchain?
  - no definition of an "abstract blockchain"
- Does Nakamoto's protocol achieve **CONSISTENCY**?
  - "Specific attacks" don't work [N'08,GKL'15, SZ'15]
  - 49.1% attack (with 10s network delays) claimed [DW'14]
- Is Nakamoto's consensus **OPTIMAL**?
  - Several issues known (load,latency,incentives)

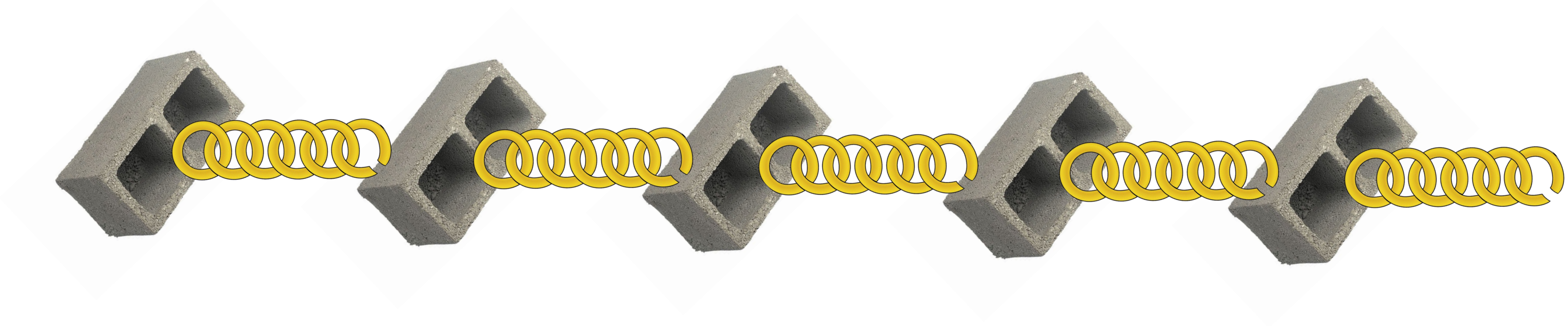
This talk

1. Desiderata of blockchain
2. Nakamoto Achieves Desiderata
3. Overcoming Bottlenecks

This talk

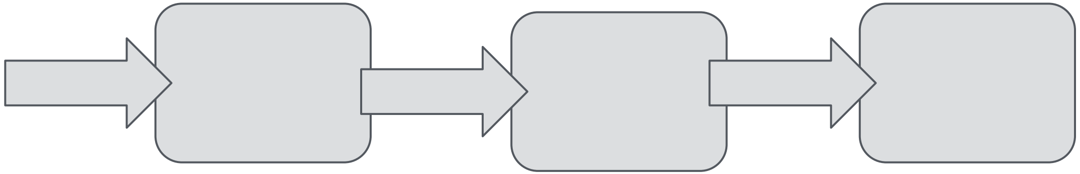
1. Desiderata of blockchain
2. Nakamoto Achieves Desiderata
3. Overcoming Bottlenecks

What is a blockchain?

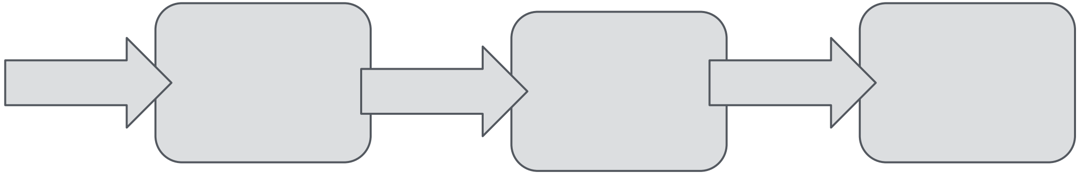


Idea: Use Proof-of-Work Puzzles to defend against sybil attacks

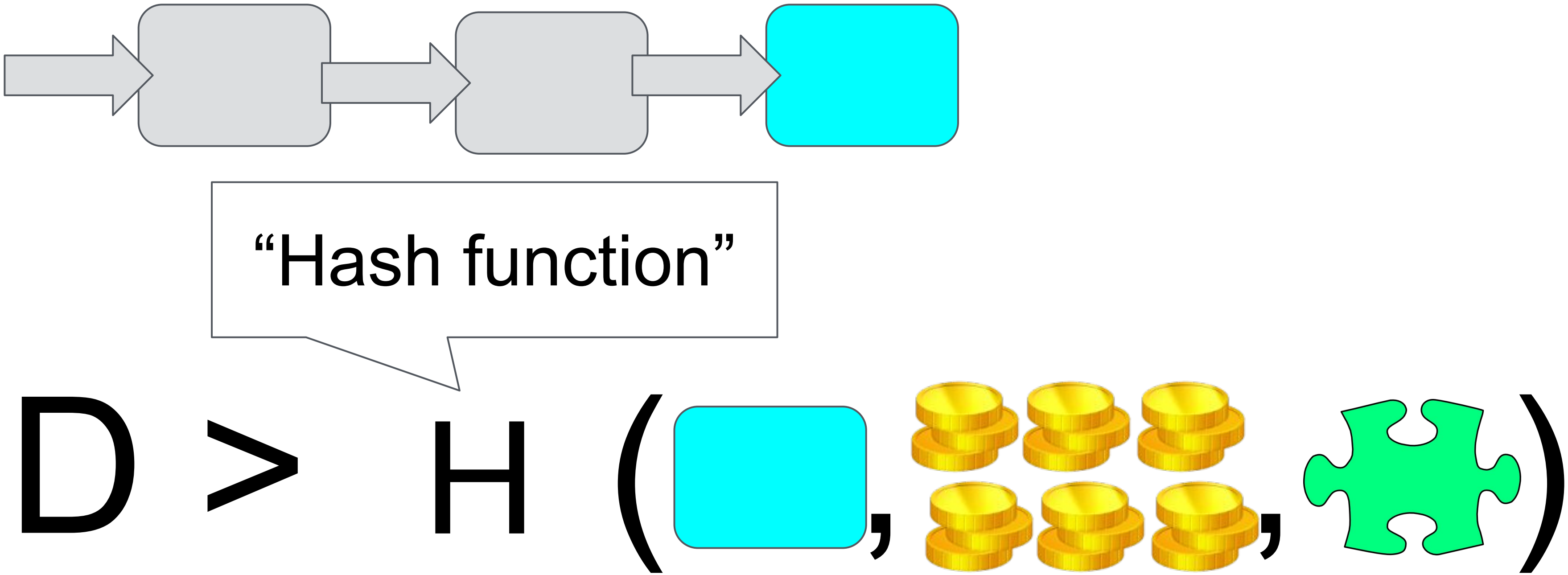
Users have to do work to cast votes.



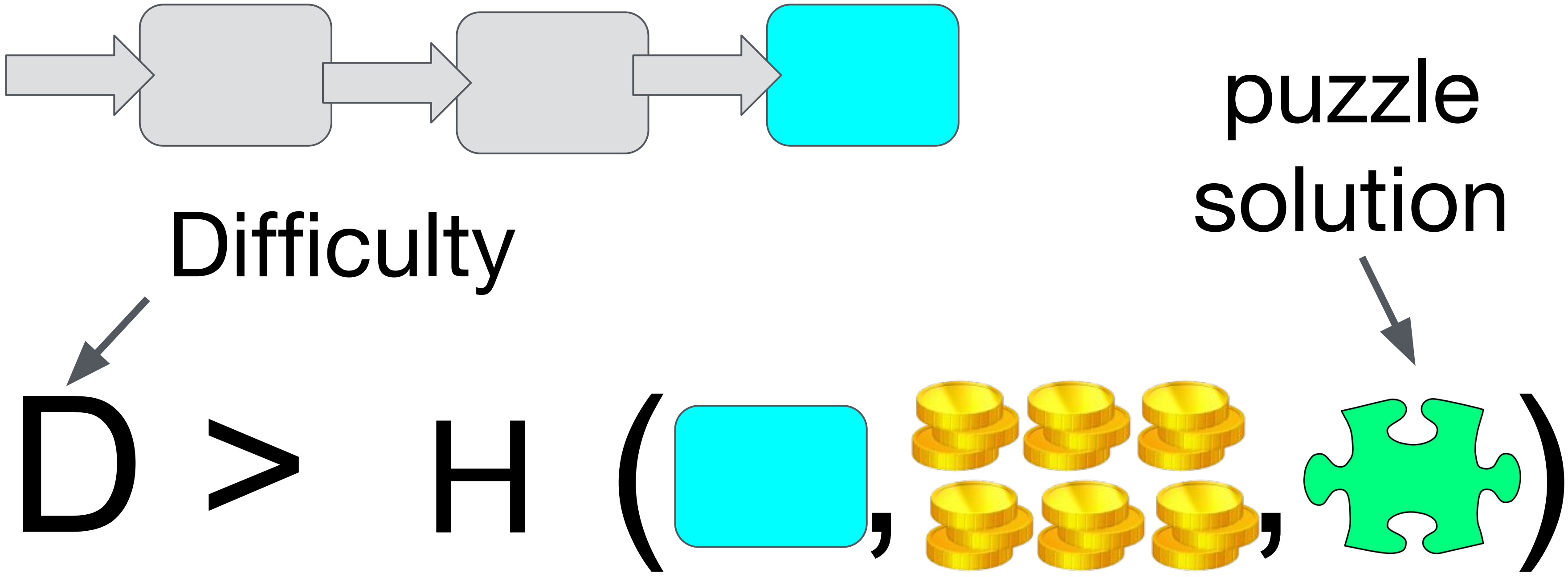
How to build a “blockchain”



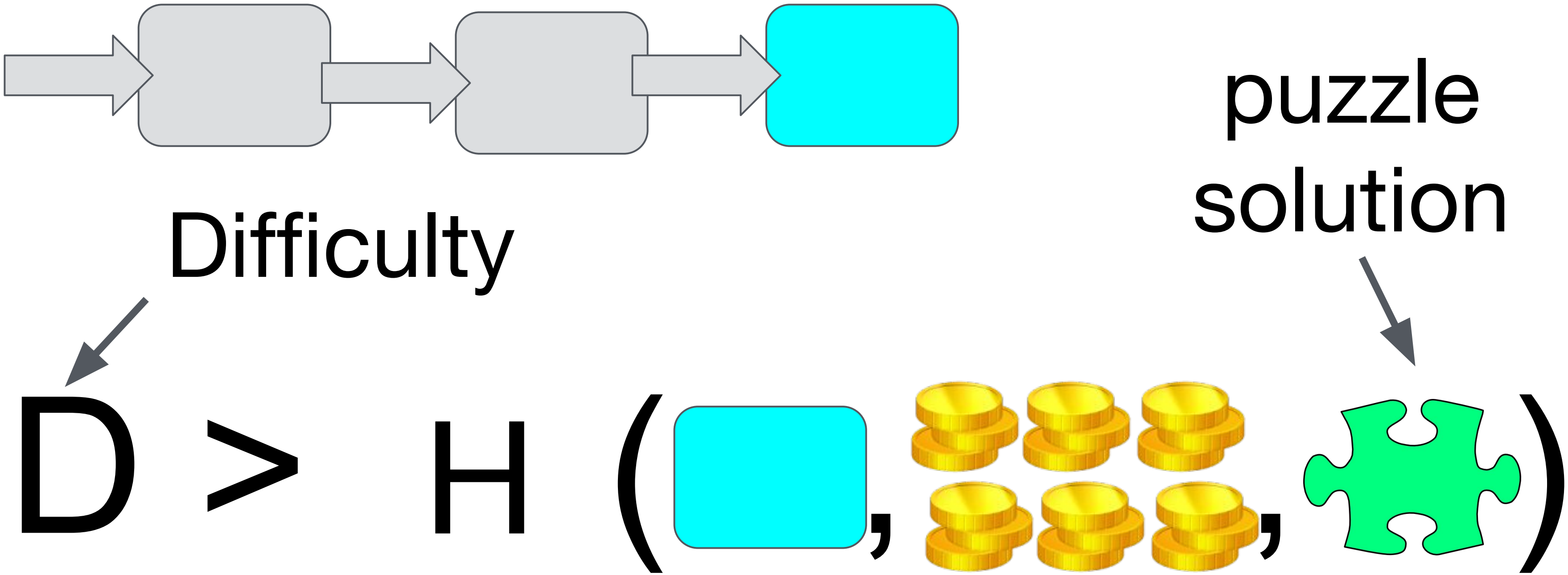
How to build a “blockchain”



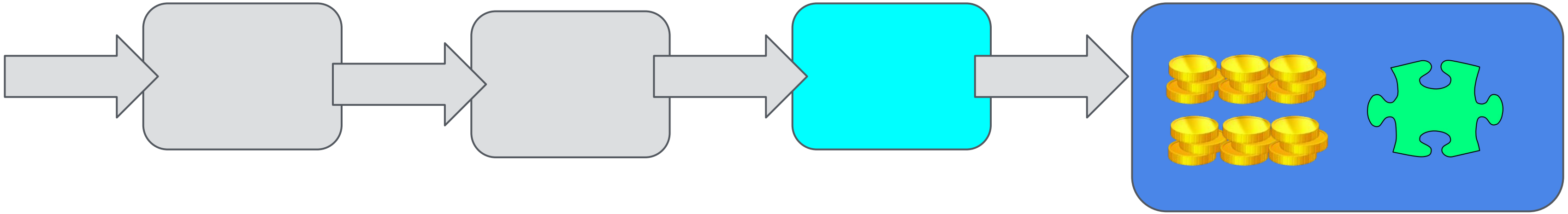
How to build a “blockchain”

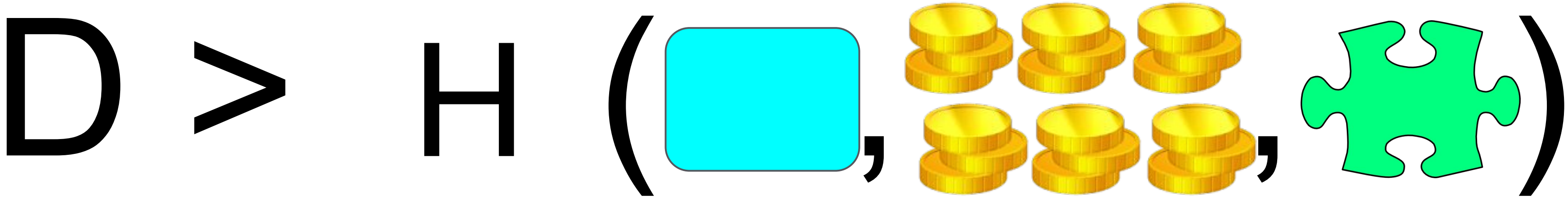


Search for a puzzle solution

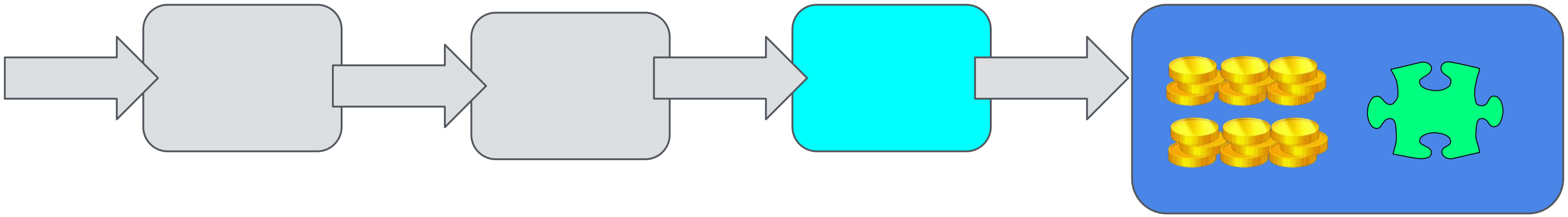


Search for a puzzle solution



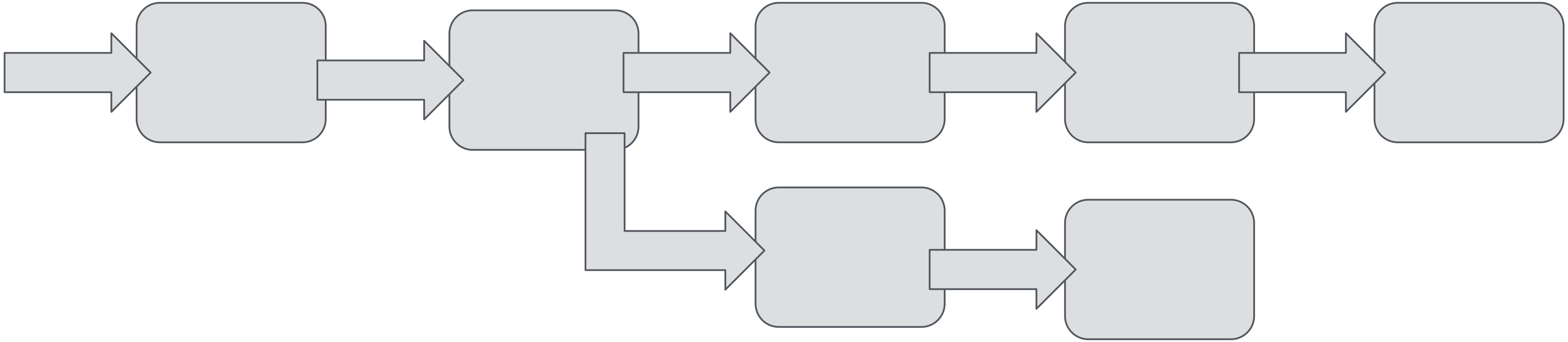


We found a new block

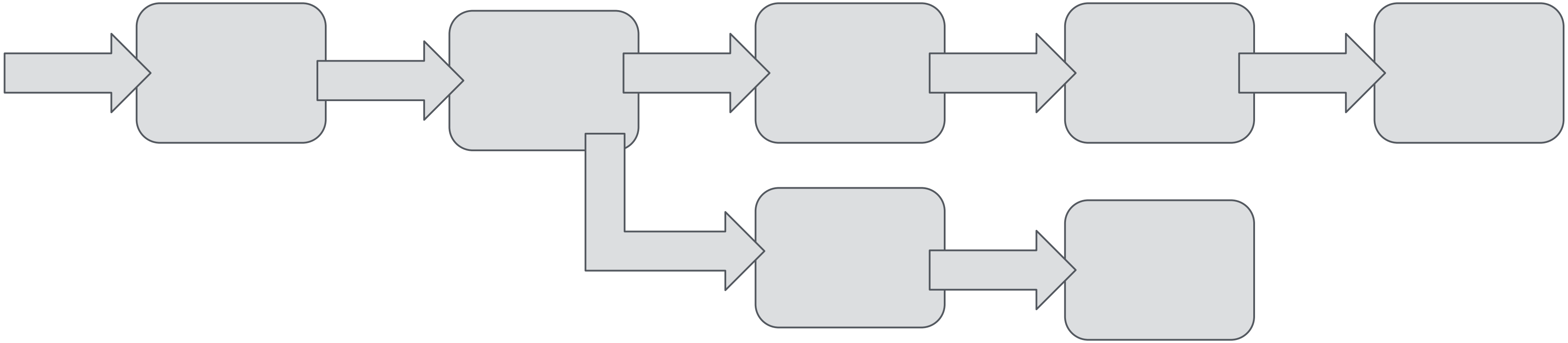


D > H ()

Best way to find a solution is brute-force search: model H as RO

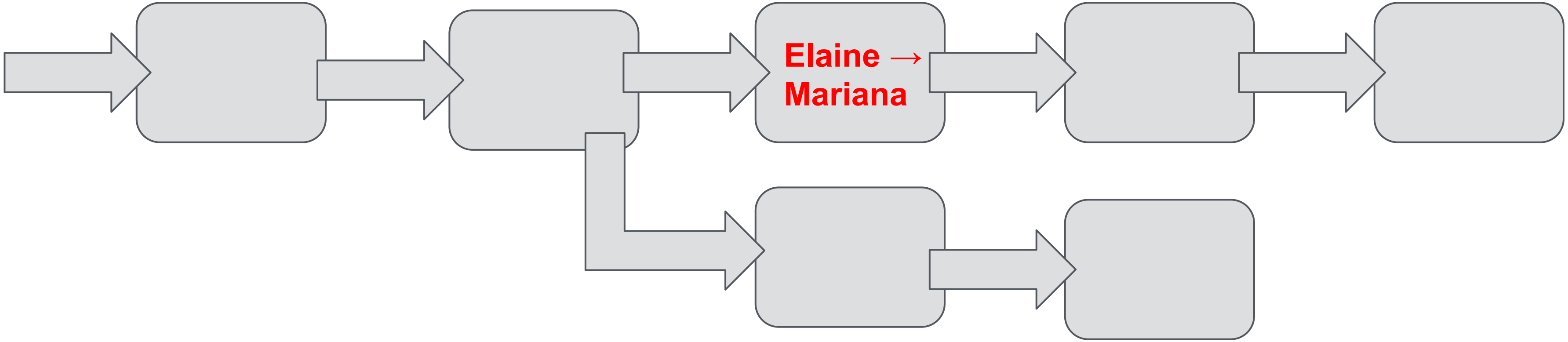


What if you join network and you see this.

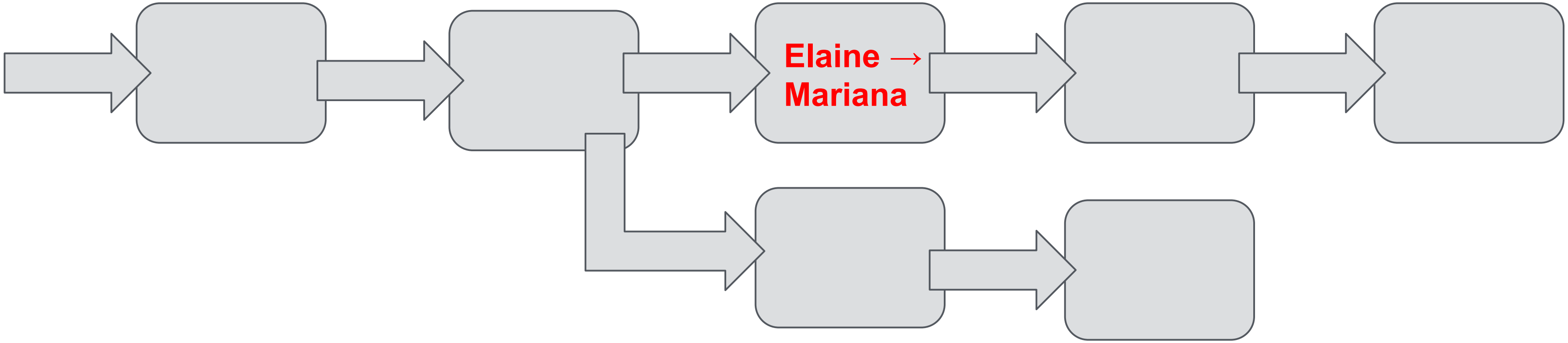


Honest nodes only “believe”

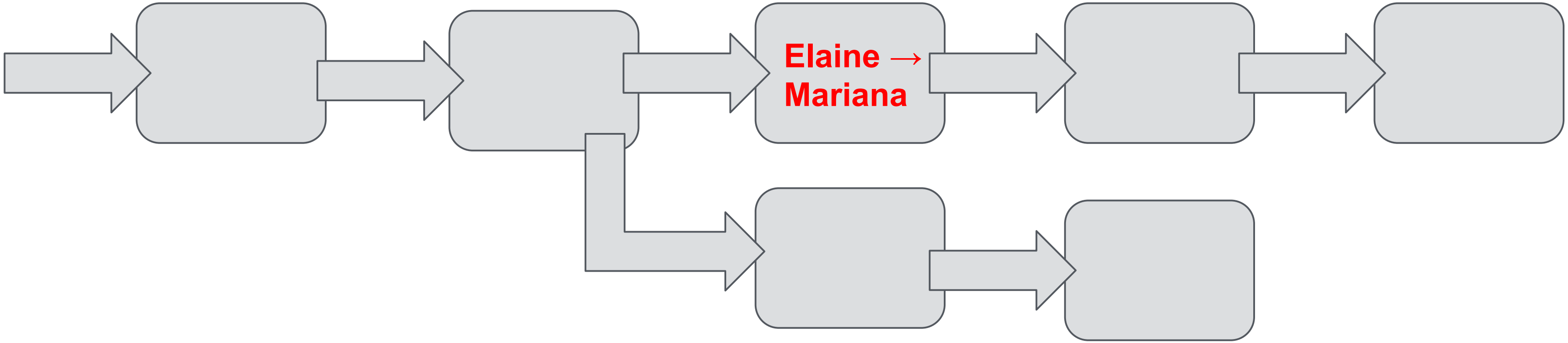
longest chain



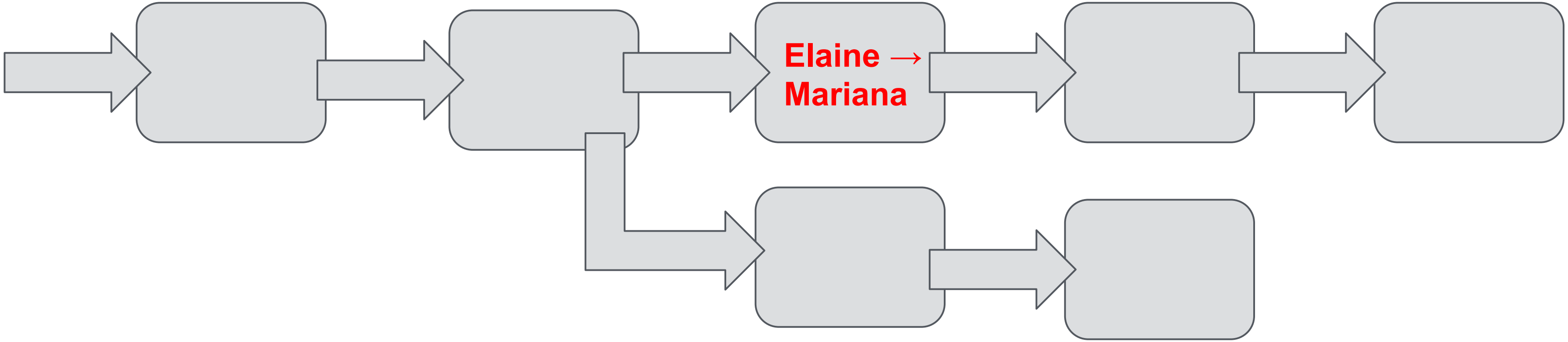
Elaine wants to erase this transaction



For Elaine to erase his transaction, he has to find a longer chain!



“If transaction is sufficiently deep, he cannot do this unless he has majority hashpower”



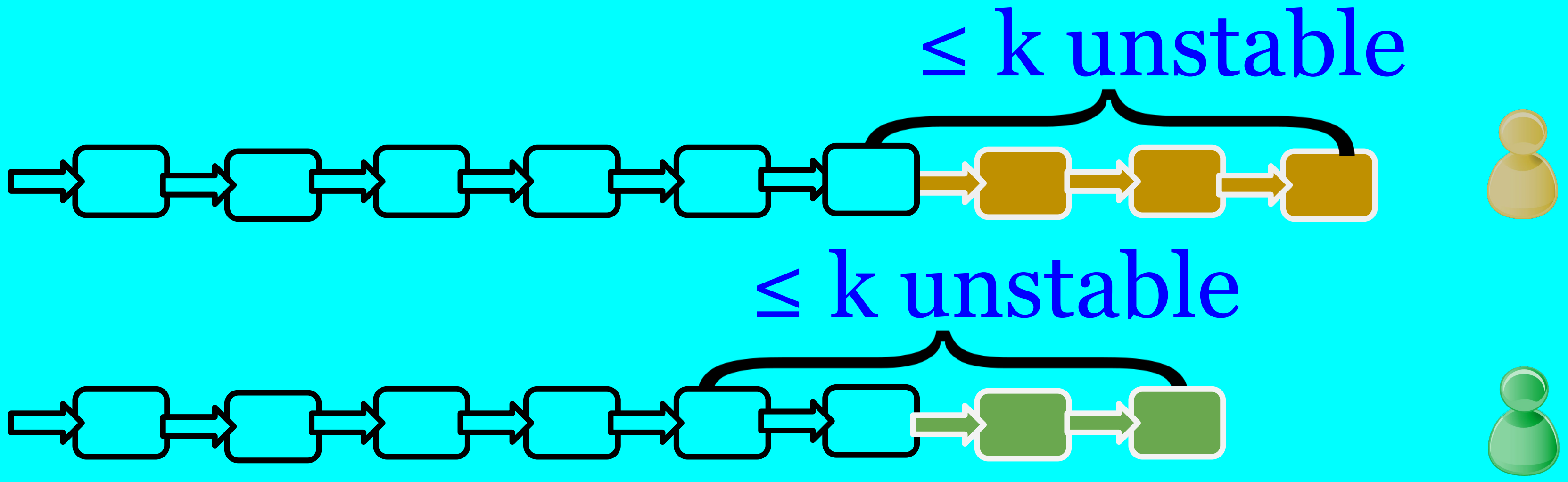
"If transaction is sufficiently deep, he cannot do this unless he has majority hashpower"

- [Nak'08]: "simply trying to mine alternative chain fails"
- [GLK'15]: in synchronous network
- [SZ'15]: "non-withholding attacks" fail also with Δ-delays

Blockchain abstraction

w/ prob exp(-k)

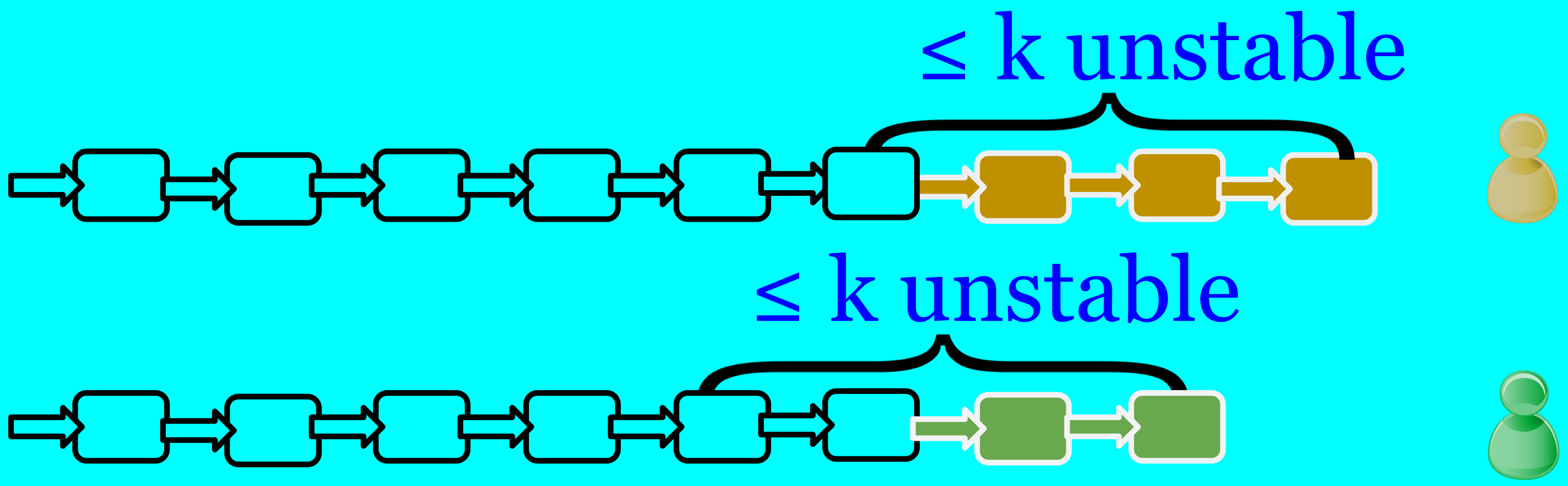
1. Consistency: Honest nodes agree on all but last k blocks



Blockchain abstra… w/ prob $exp(-k)$

Future-self consistency

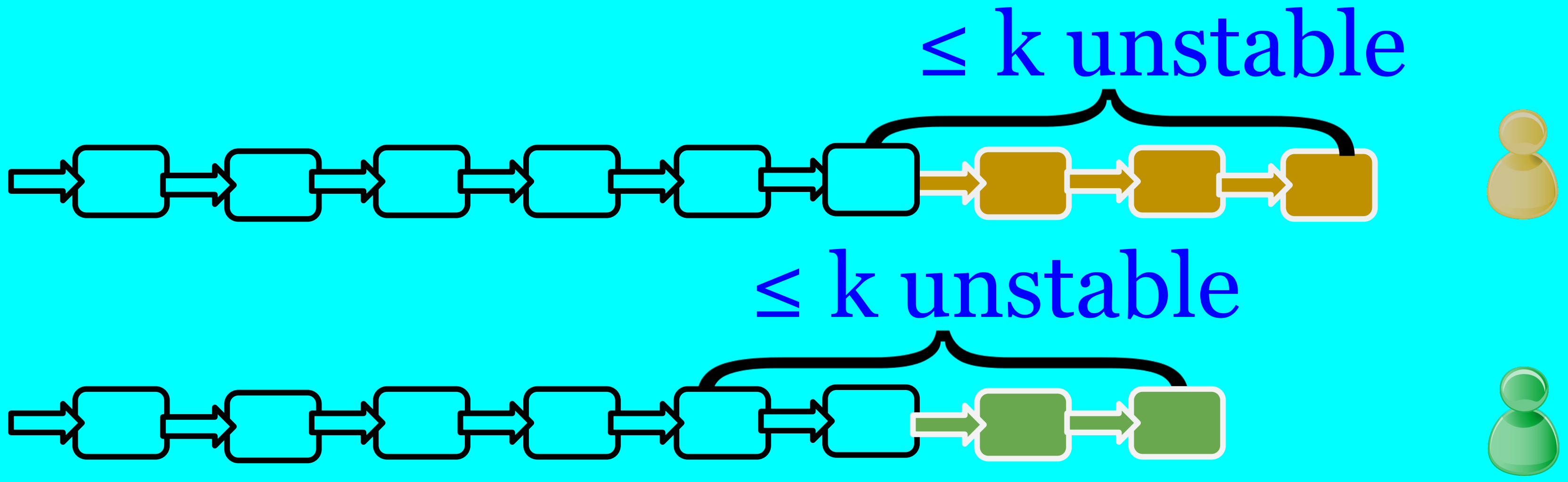
1. Consistency: Honest nodes agree on all but last k blocks



Blockchain abstraction

w/ prob exp(-k)

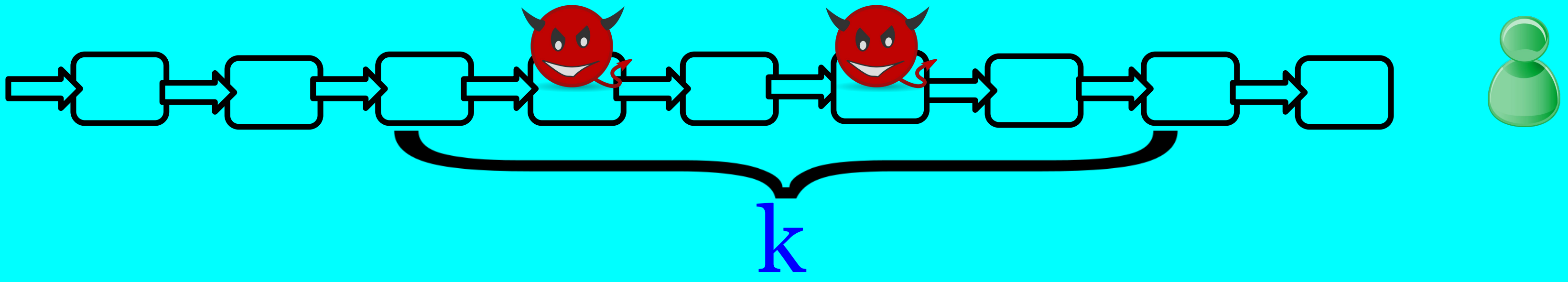
1. **Consistency**: Honest nodes agree on all but last k blocks



Blockchain abstraction

w/ prob exp(-k)

1. Consistency: Honest nodes agree on all but last k blocks
2. Chain quality: Any consecutive k blocks contain “sufficiently many” honest blocks



Blockchain abstraction

w/ prob $\exp(-k)$

1. Consistency: Honest nodes agree on all but last k blocks
2. Chain quality: Any consecutive k blocks contain "sufficiently many" honest blocks
3. Chain growth: Chain grows at a steady rate

Blockchain implies "state machine replication" in the permissionless model

1. Consistency
2. Chain quality
3. Chain growth

Traditional

"state machine replication"

1. Consistency
2. Liveness

This talk

1. Desiderata of blockchain
2. Nakamoto Achieves Desiderata
3. Overcoming Bottlenecks

Theorem [P-Seeman-Shelat]:

For every $\rho<1/2$, if "mining difficulty" is appropriately set (as a function of the network delay Δ, and total mining power), Nakamoto's blockchain guarantees:

- Consistency
- Chain quality: $1 - \rho/(1-\rho)$
- Chain growth: $O(1/\Delta)$

where ρ adv's fraction of hashpower, and **adv controls the network**

Theorem [P-Seeman-Shelat]:

For every $\rho<1/3$, if "mining difficulty" is appropriately set (as a function of the network delay Δ, and total mining power), Nakamoto's blockchain guarantees:

- Consistency
- Chain quality: $1 - (1/3)/(2/3) = 1/2$
- Chain growth: $O(1/\Delta)$

where ρ adv's fraction of hashpower, and **adv controls the network**

Theorem [P-Seeman-Shelat]:

For every $\rho<1/2$, if "mining difficulty" is appropriately set (as a function of the network delay Δ, and total mining power), Nakamoto's blockchain guarantees:

- Consistency
- Chain quality: $1 - \rho/(1-\rho)$
- Chain growth: $O(1/\Delta)$

where ρ adv's fraction of hashpower, and **adv controls the network**

Theorem [P-Seeman-Shelat]:

For every $\rho<1/2$, if "mining difficulty" is appropriately set (as a function of the network delay Δ, and total mining power), Nakamoto's blockchain guarantees:

- Consistency
- Chain quality: $1 - \rho/(1-\rho)$
- Chain growth: $O(1/\Delta)$

"Blocks are found SLOWER than Δ"

where ρ adv's fraction of hashpower, and **adv controls the network**

Theorem [P-Seeman-Shelat]:

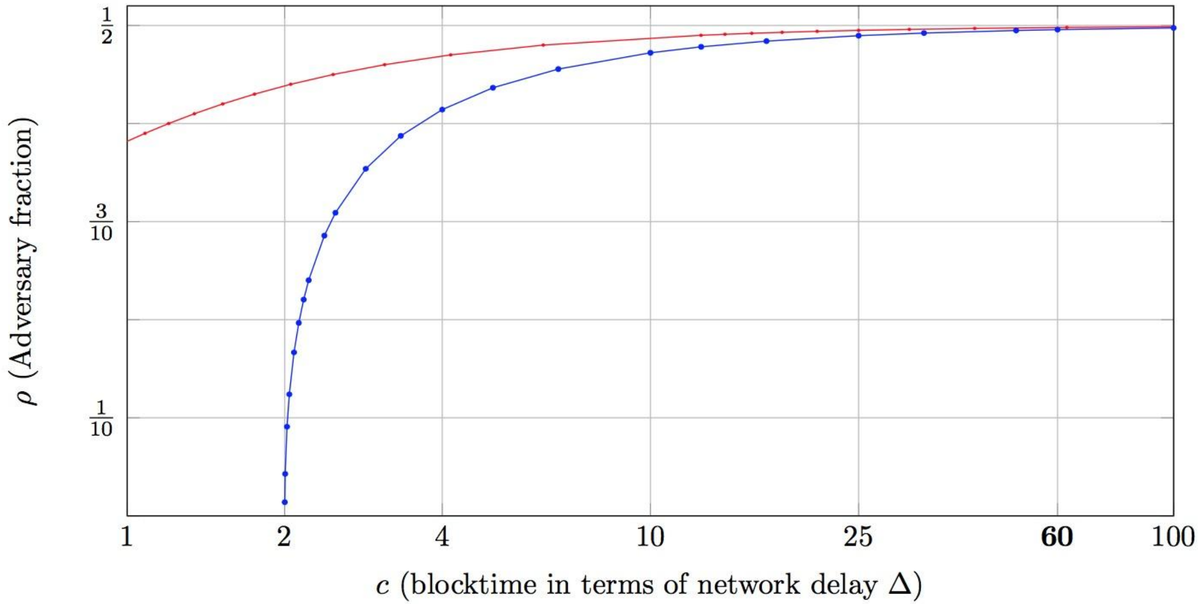
For every $\rho<1/2$, if “mining difficulty” is appropriately set (as a function of the network delay Δ, and total mining power), Nakamoto’s blockchain guarantees:

- Consistency
- Chain quality: $1 - \rho/(1-\rho)$
- Chain growth: $O(1/\Delta)$

“Blocktime” >> Δ

where ρ adv’s fraction of hashpower, and **adv controls the network**

"Appropriately set"



When c = 60 (10 min blocktime, 10s network delays)

Secure: $\rho < 49.57$ (contradicts [DW'14]'attack!)

Attack: $\rho > 49.79$

“Appropriately set”

$$\alpha(1 - 2(\Delta + 1)\alpha) > \beta.$$

Mining rate of honest players

Network Delay

Mining rate of Adv

Theorem [Security of Nakamoto]

For every $\rho<1/2$, if mining difficulty is appropriately set (as a function of the network delay, and total mining power), Nakamoto's blockchain guarantees a) consistency, b) chain quality $1 - \rho/(1-\rho)$, and c) Chain growth: $O(1/\Delta)$

Theorem [Blatant attack]:

For every $\rho>0$, for every mining difficulty, there exists a network delay such that Nakamoto's blockchain is inconsistent and has 0 chain quality

This talk

1. Desiderata of blockchain
2. Nakamoto Achieves Desiderata
3. Overcoming Bottlenecks

Nakamoto: ISSUES

Terrible performance

Not incentive compatible

Bitcoin has terrible performance

- Cost per confirmed transaction in Bitcoin: **$6.20**

- **7 tx/sec**, **10 min** TX confirmation time

c.f. Visa credit card: average **2,000 tx/sec**, peak **59,000 tx/sec**

[Source: K. Croman et al. On Scaling Decentralized Blockchains. In Bitcoin workshop, 2016.]

Traditional BFT protocols are performant

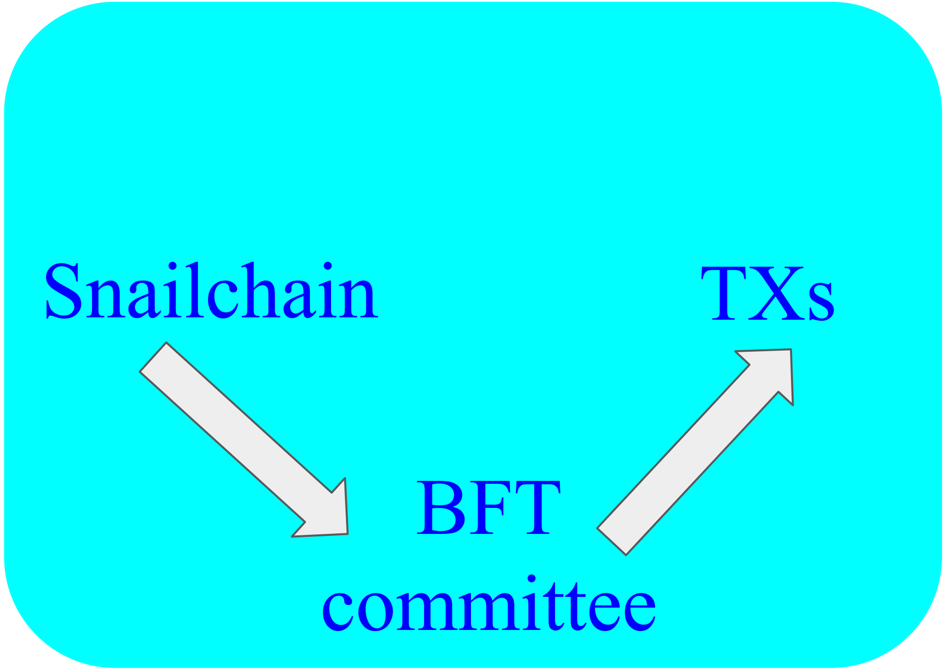
PBFT at ~100 nodes:

Throughput: **~10,000 tx/sec**

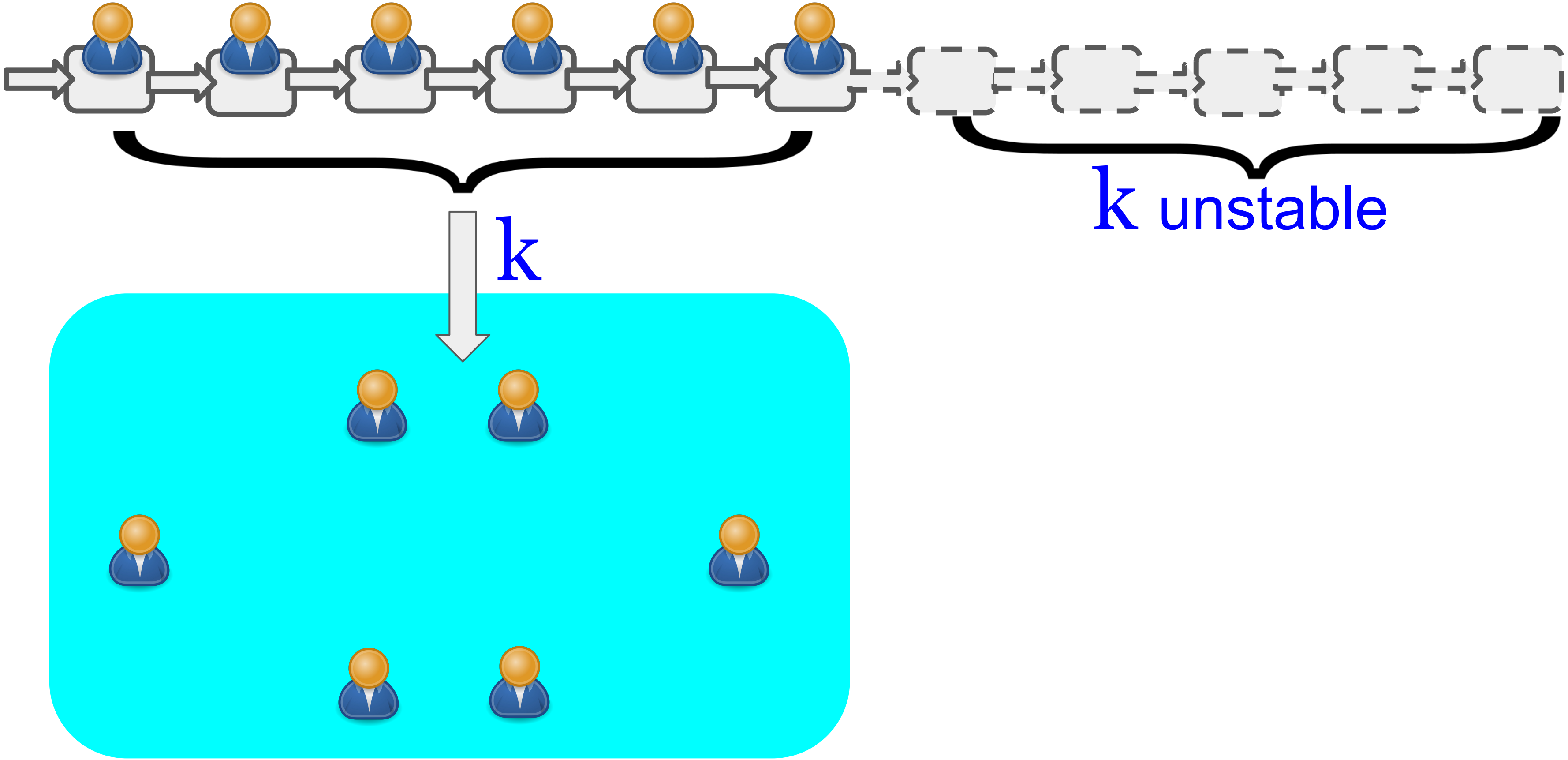
Confirmation time: **~ seconds**

[Source: K. Croman et al. On Scaling Decentralized Blockchains. In Bitcoin workshop, 2016.]

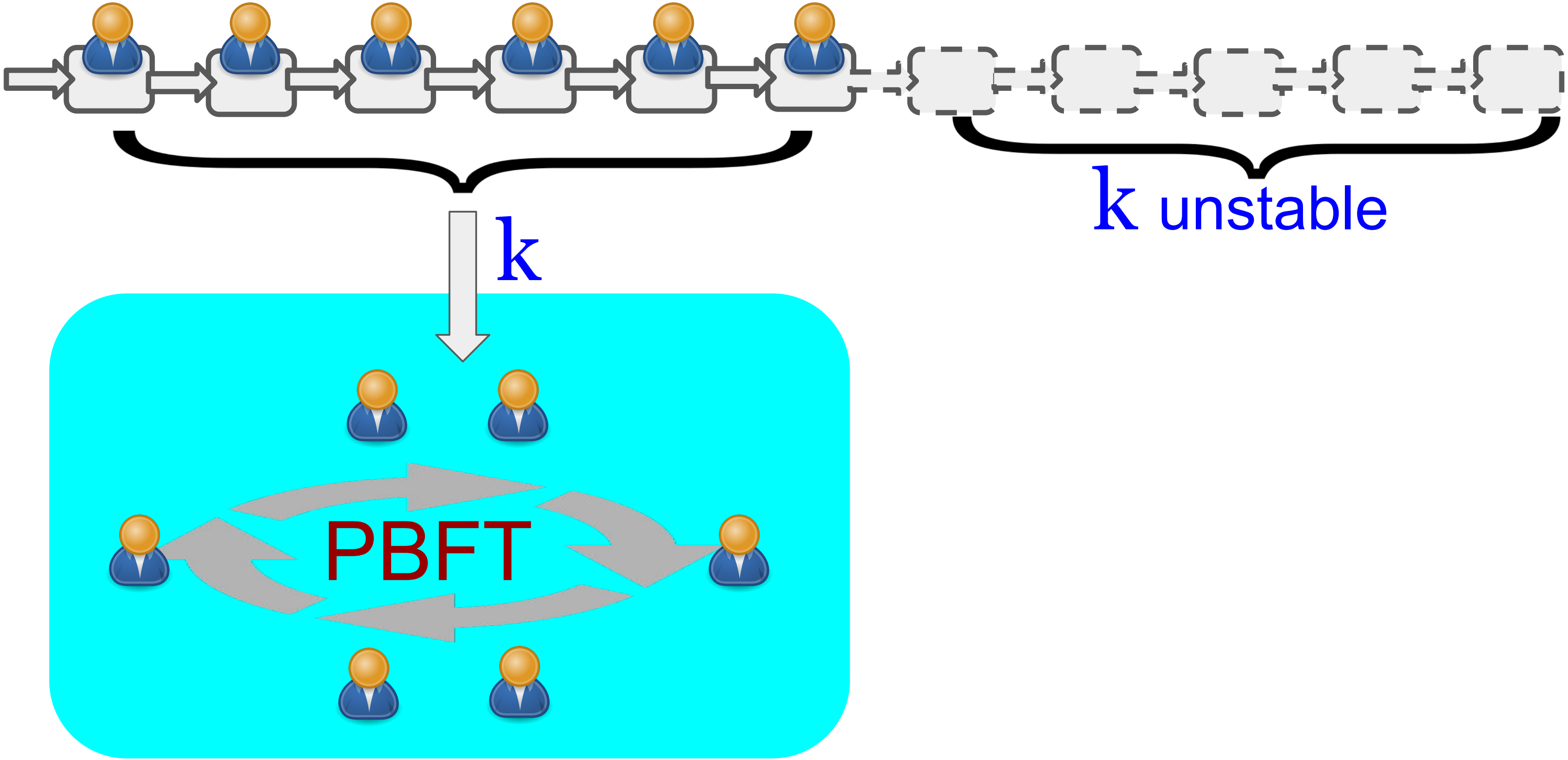
Hybrid consensus [P-Shi]



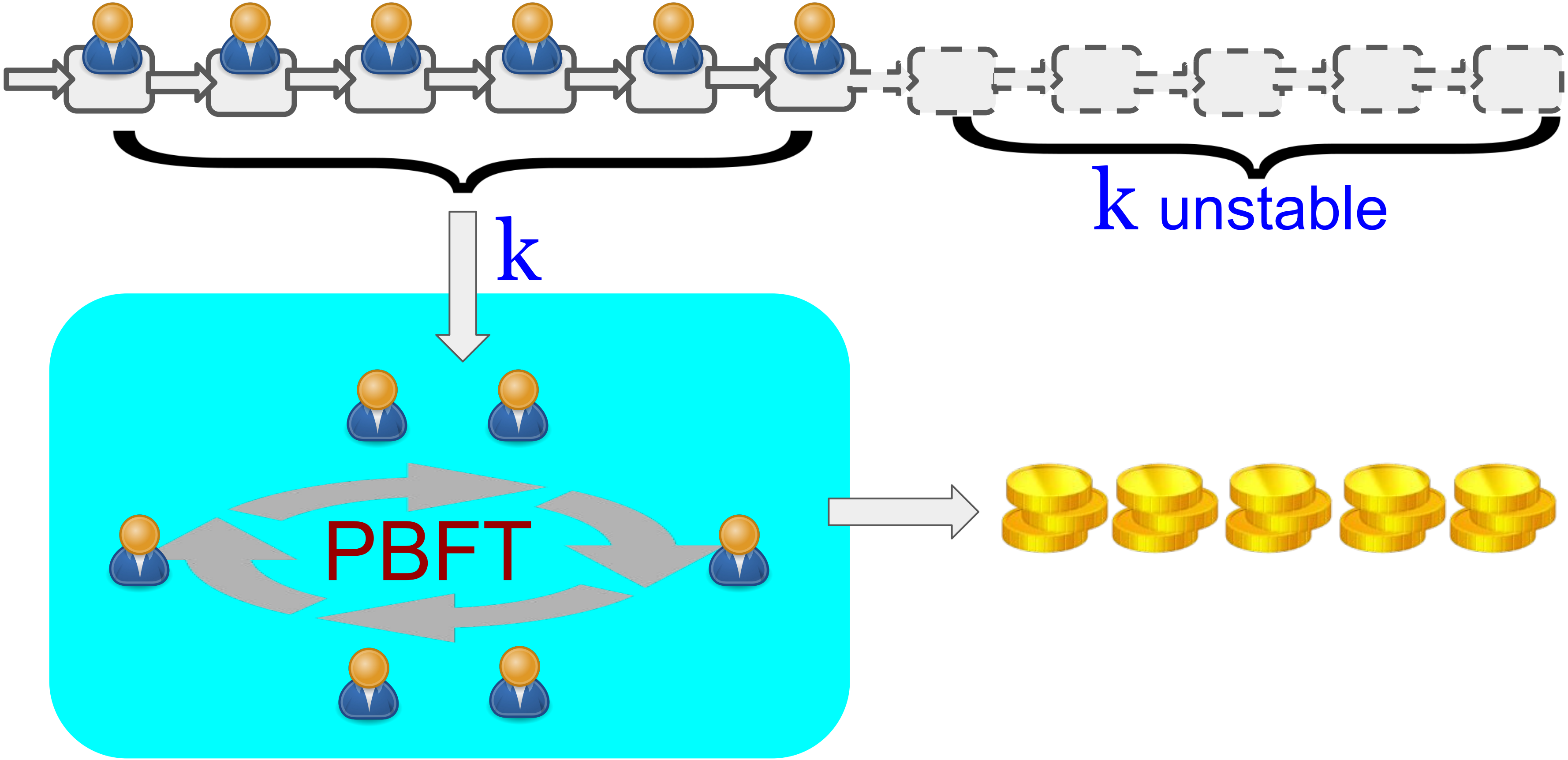
Hybrid Consensus: The idea



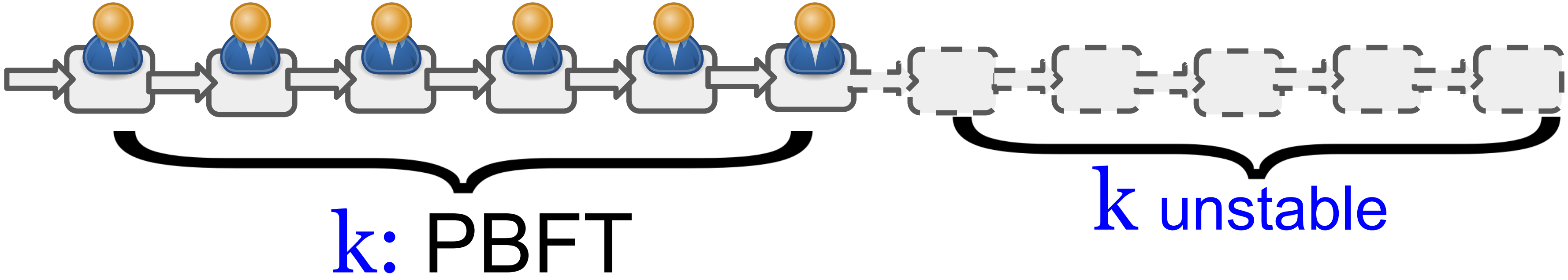
Hybrid Consensus: The idea



Hybrid Consensus: The idea



Hybrid Consensus: The idea

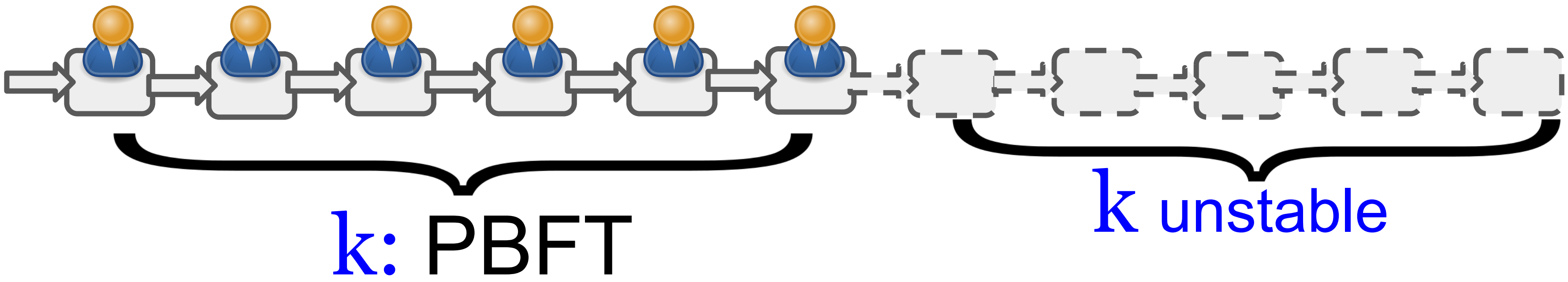


Chain quality: ⅔ committee honest (if ¾ honest overall)

Chain growth: this won’t take too long

Consistency: everyone agrees on committee

Hybrid Consensus: The idea



Achieves static security

Not adaptively secure

- Can deal with it using rotating committees

Summary

- Nakamoto’s protocol achieves strong robustness properties, assuming “honest majority of computational power”
  - ➔ Assuming puzzle difficulty is appropriately set as a function of network delay Δ
  - ➔ Blocktime need to be rougly 10 * Δ for to handle $\rho > 0.45$
  - ➔ Leads to high latency (slow confirmation times)
- Can BOOTSTRAP Nakamoto into new blockchain protocols
  - ➔ Low latency (fast confirmation times)
  - ➔ incentive compatible: fruit chains