

# Distributed ledgers: how, why, and why not?

---

**Sarah Meiklejohn (University College London)**

data consumers



company





company





company

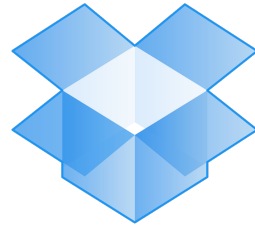




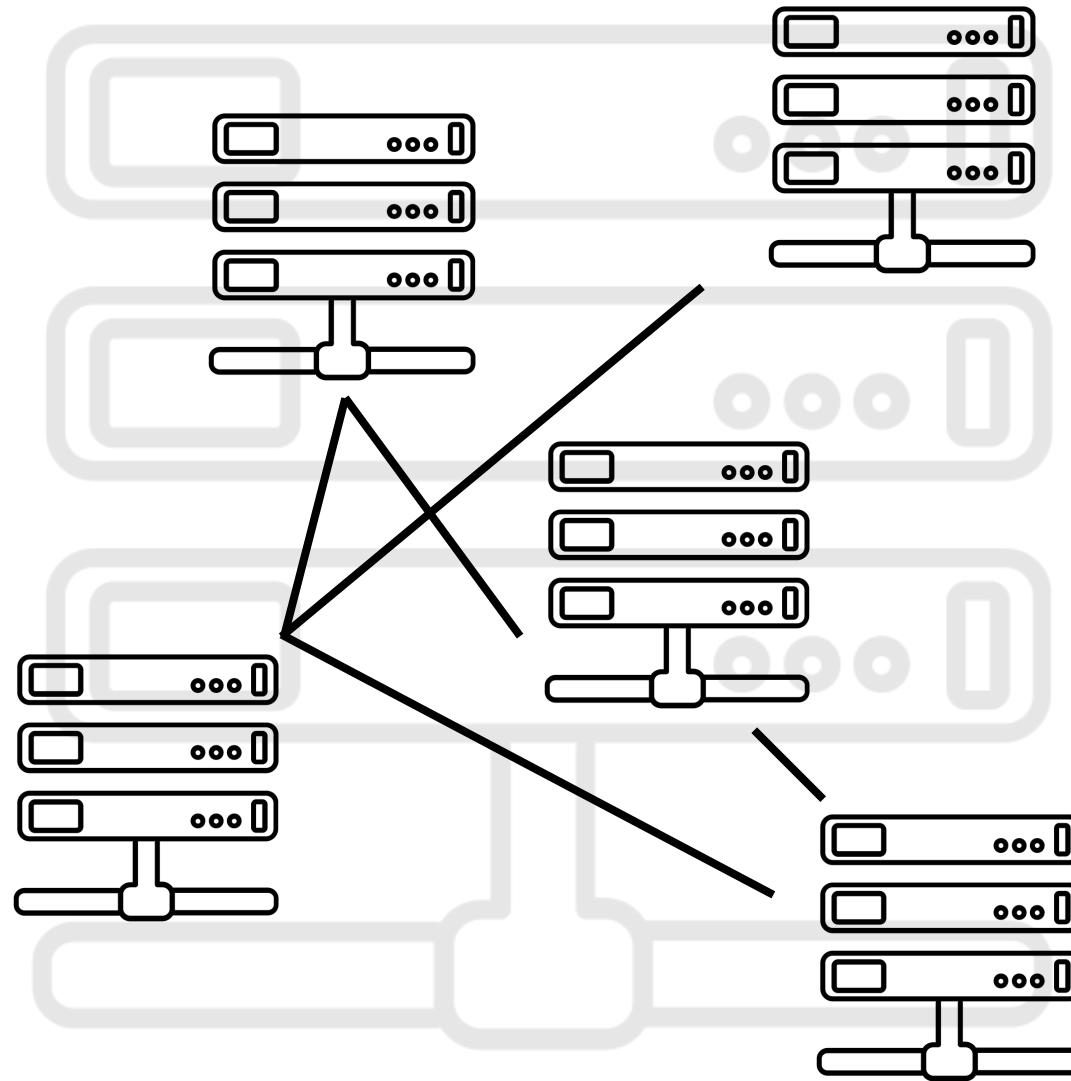
company



data producers



data consumers



data producers



# top ten obstacles for blockchains

**10 usability**

**9 governance**

**8 comparisons**

**7 key management**

**6 agility**

**5 interoperability**

**4 scalability**

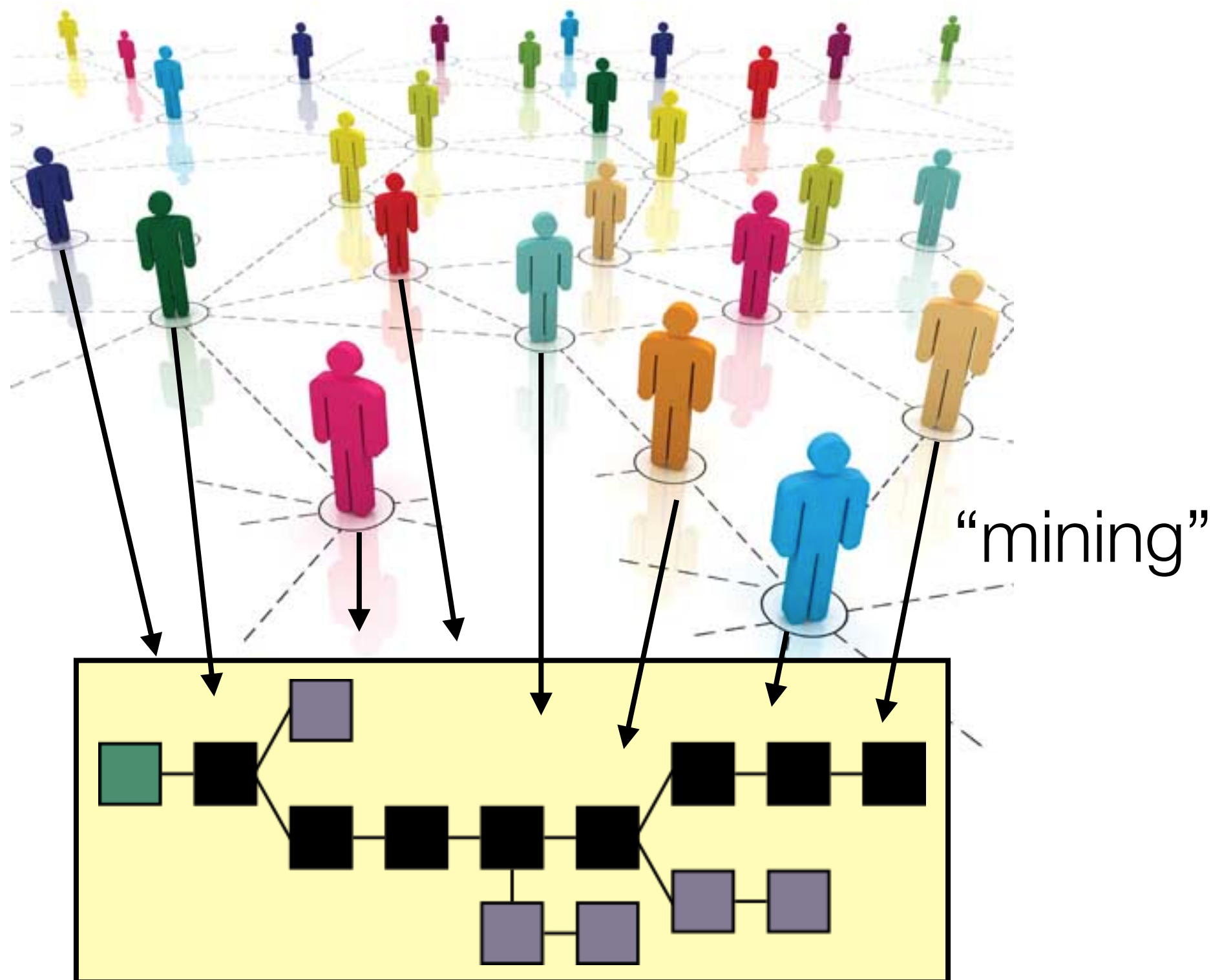
**3 cost-effectiveness**

**2 privacy**

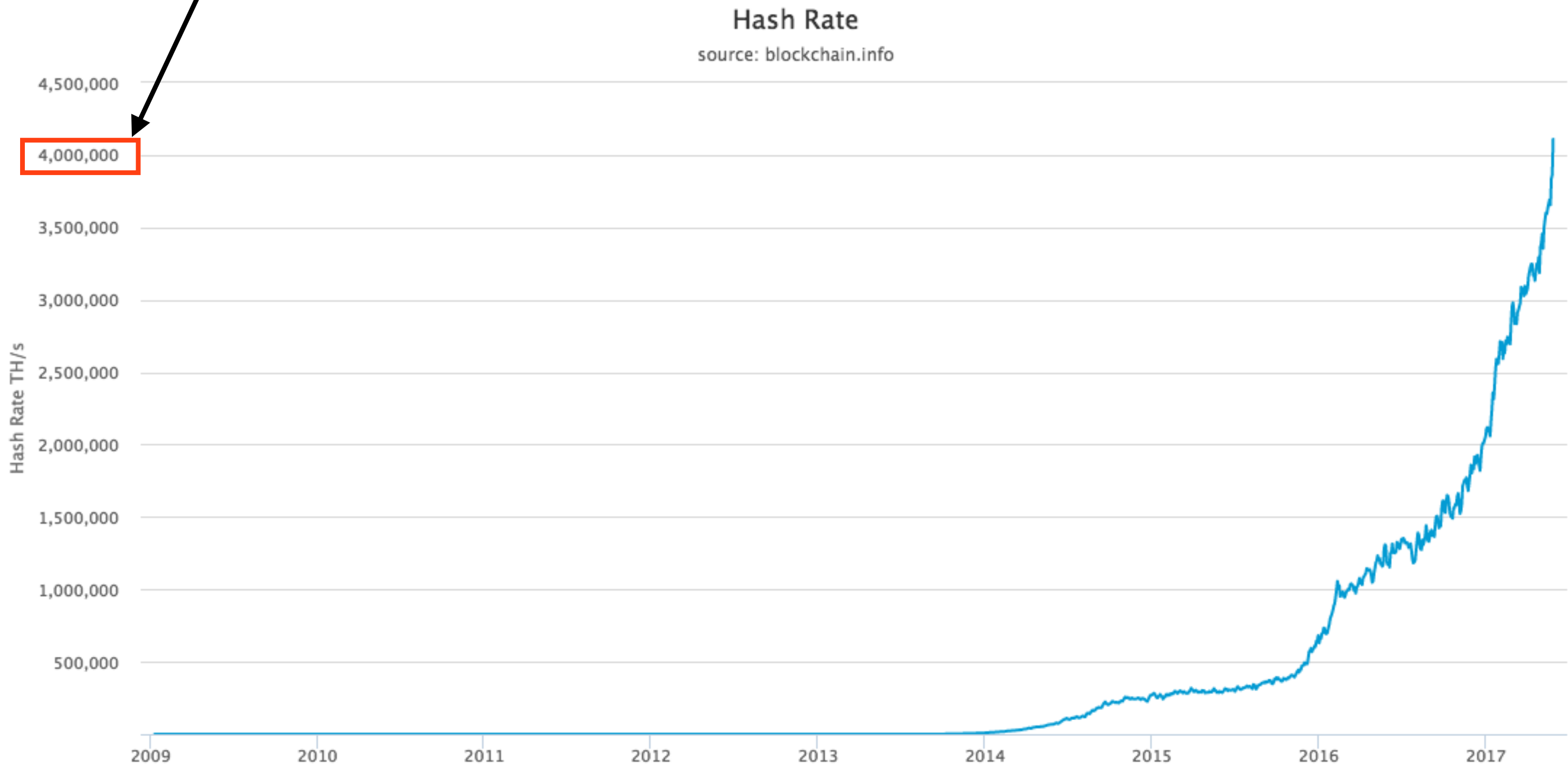
**1 scalability**

- 10 usability**
- 9 governance**
- 8 comparisons**
- 7 key management**
- 6 agility**
- 5 interoperability**
- 4 scalability**
- 3 cost-effectiveness**
- 2 privacy**
- 1 scalability**

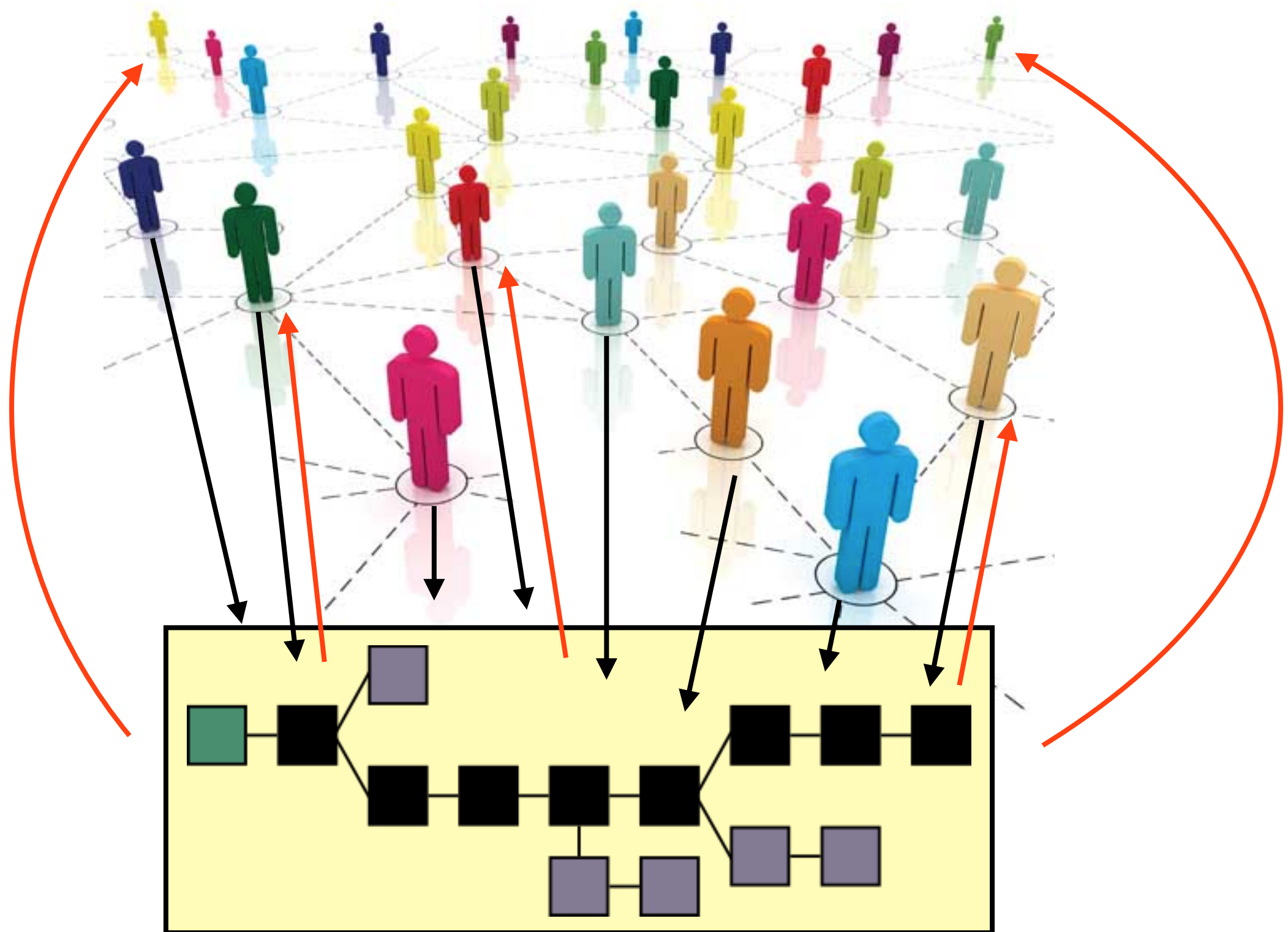
# Bitcoin / blockchains / distributed ledgers



over 4 EH/s ( $4 \times 10^{18}$  H/s) to achieve 7 tx/s!



# full state replication

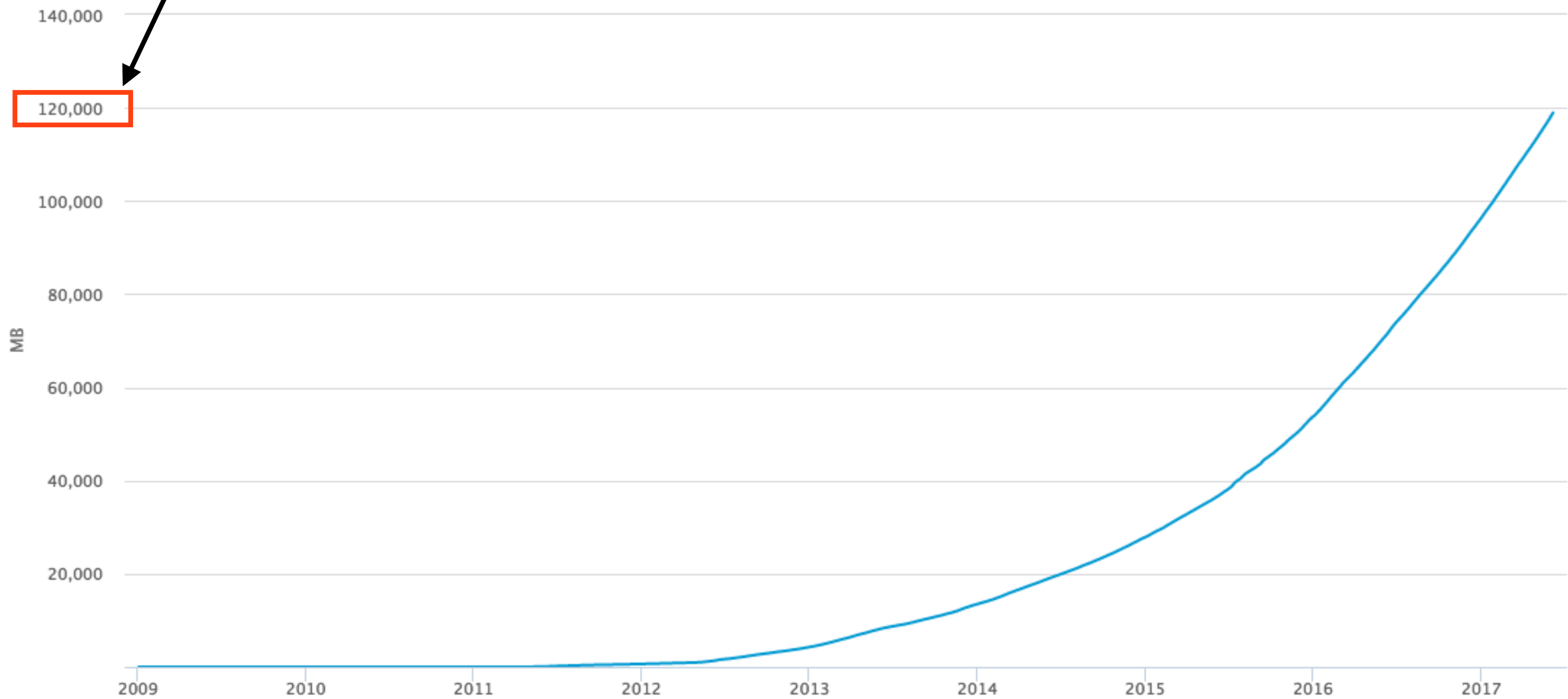




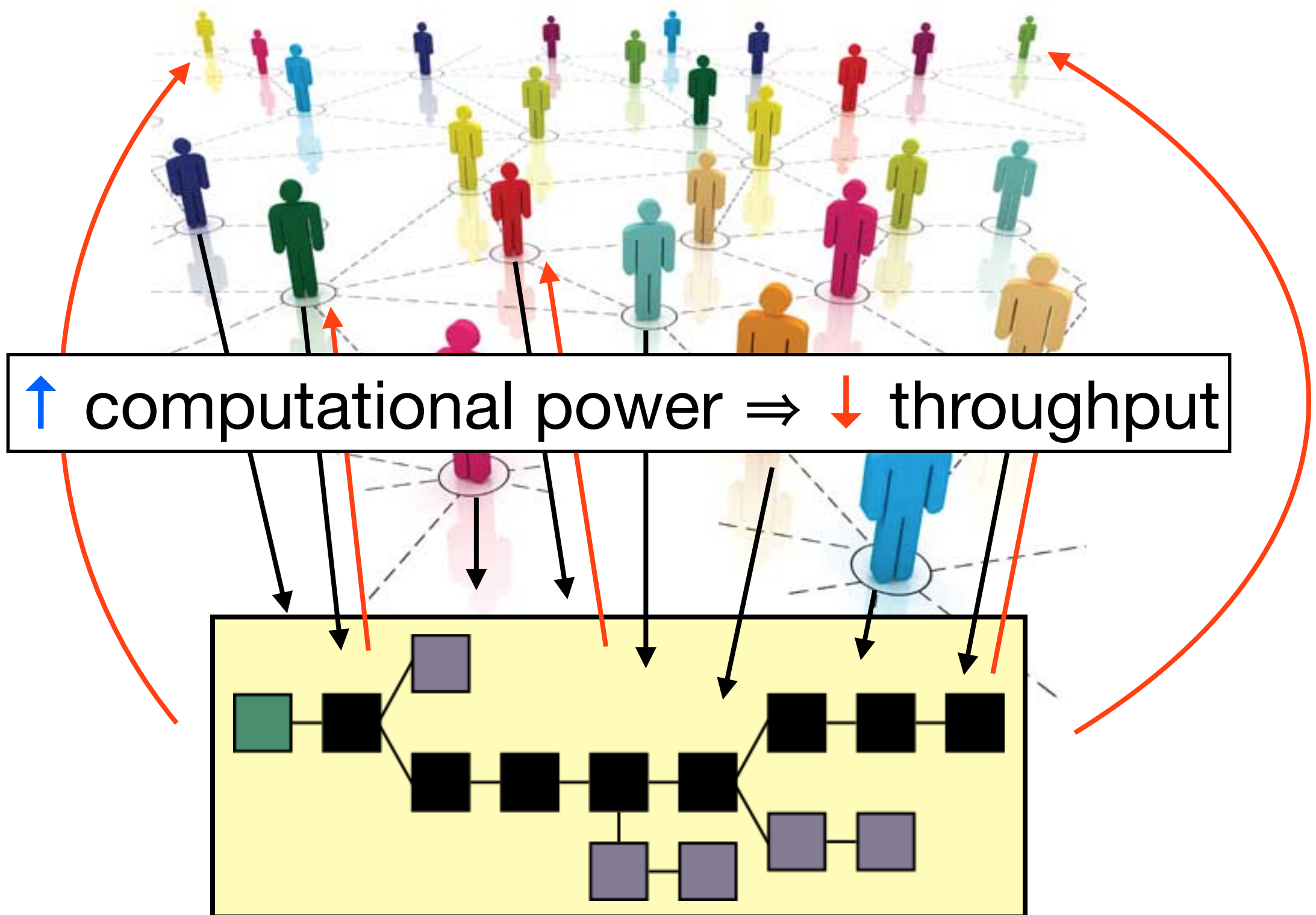
# 120 GB and (always) rising

## Blockchain Size

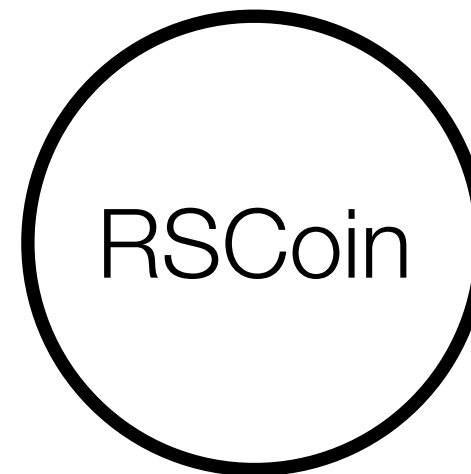
source: blockchain.info



# full state replication



# RSCoin [DM NDSS'16]



monetary supply

decentral

central

central

ledger

decentral

distribute

central

transparent?

y

y (or n)

n

pseudonyms?

y

y (or n)

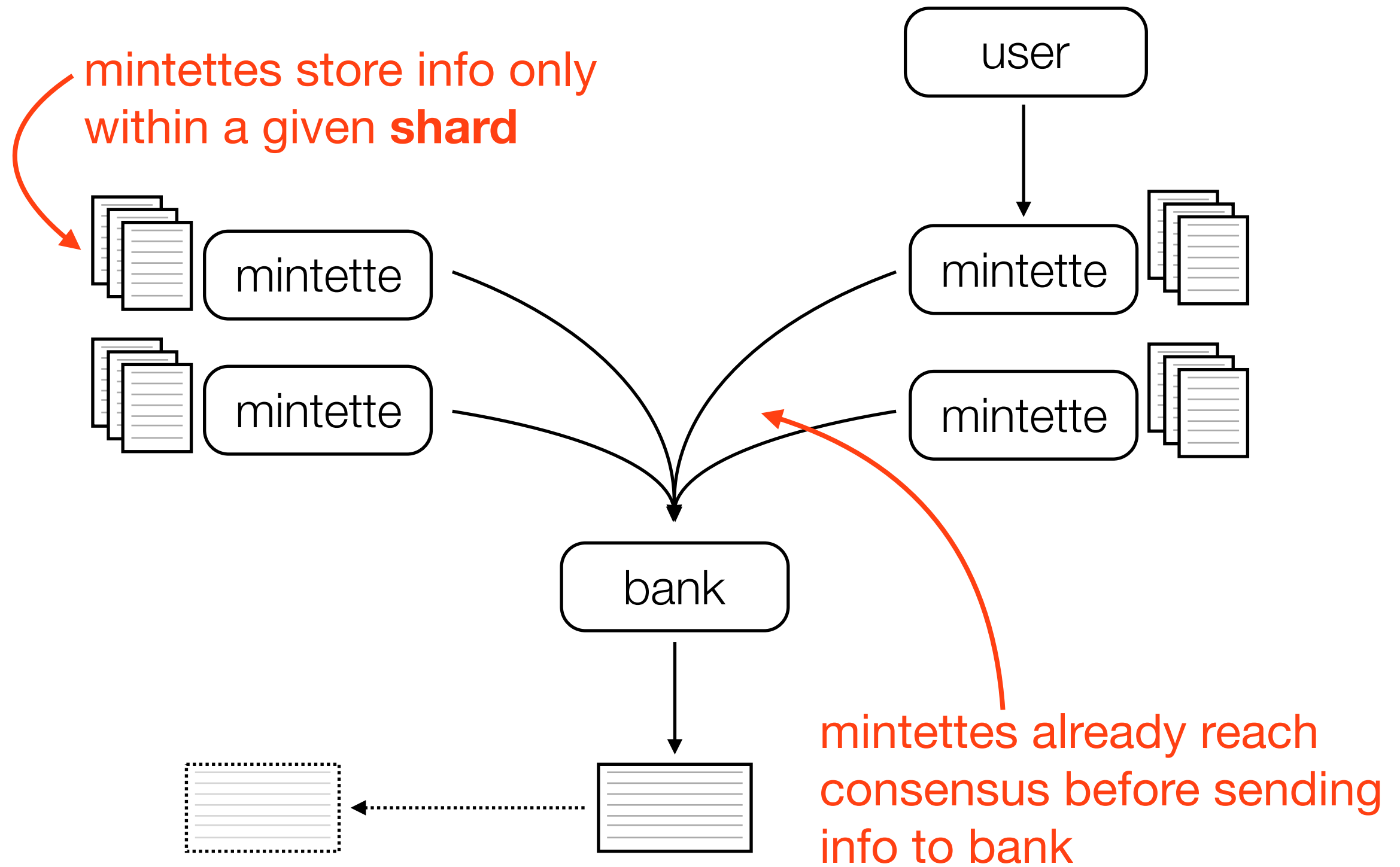
n

computation

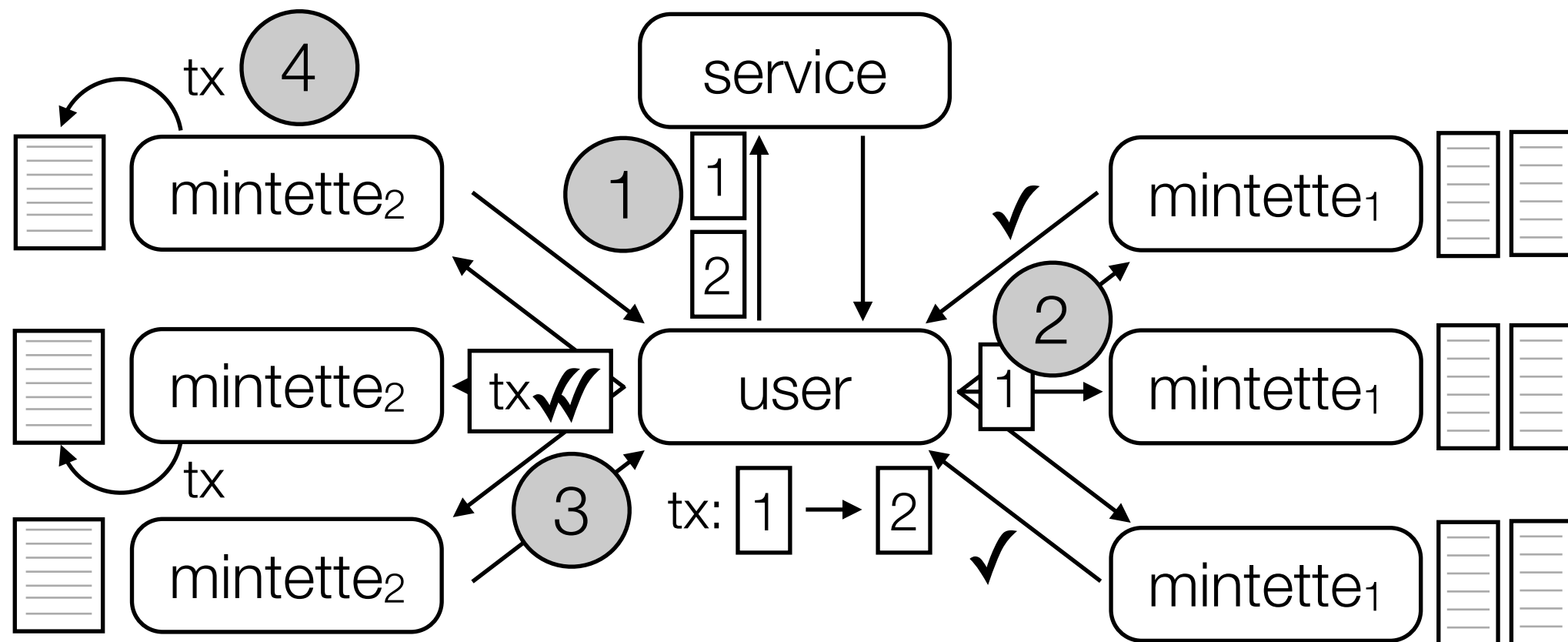
high!

low

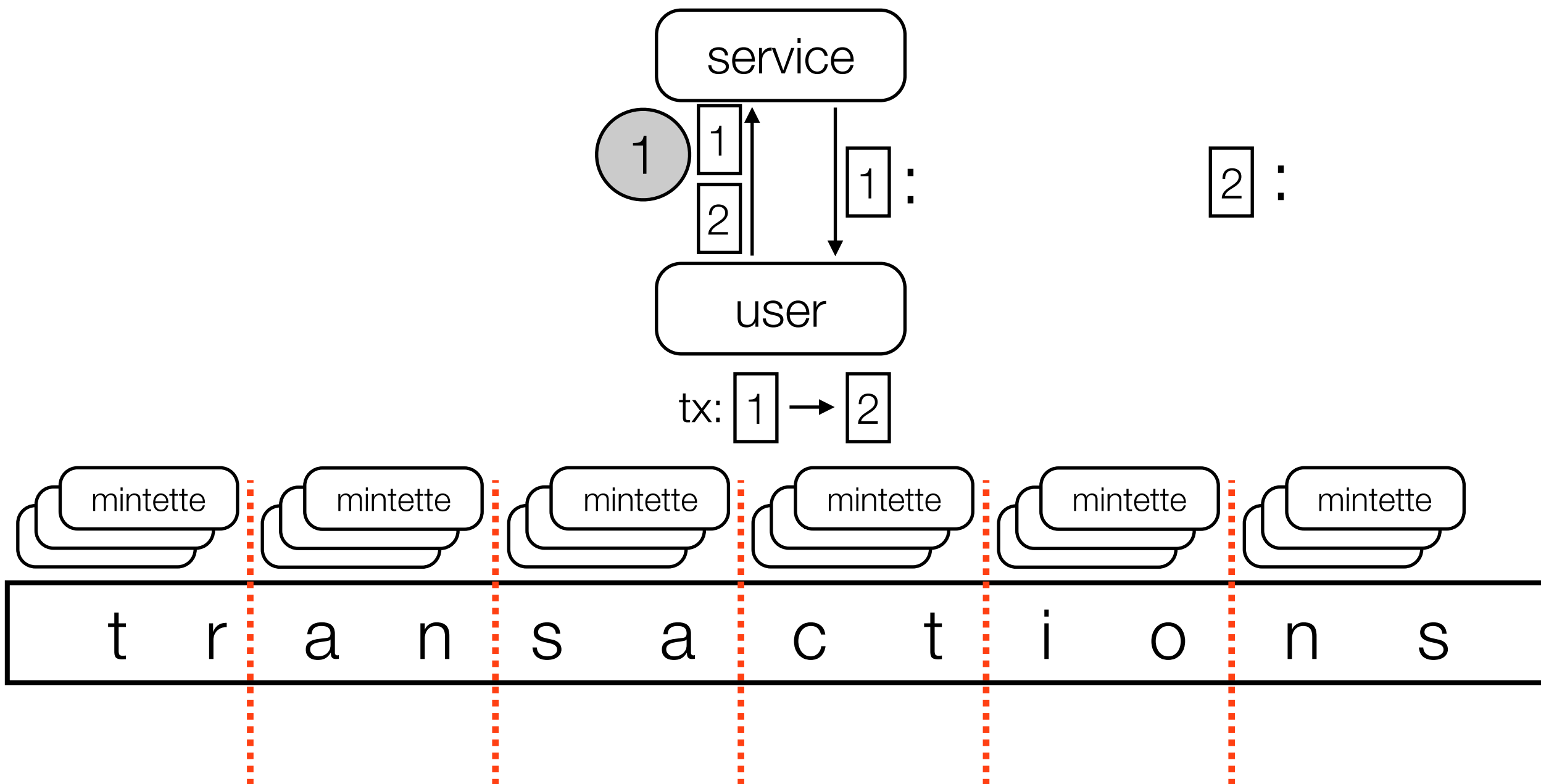
low



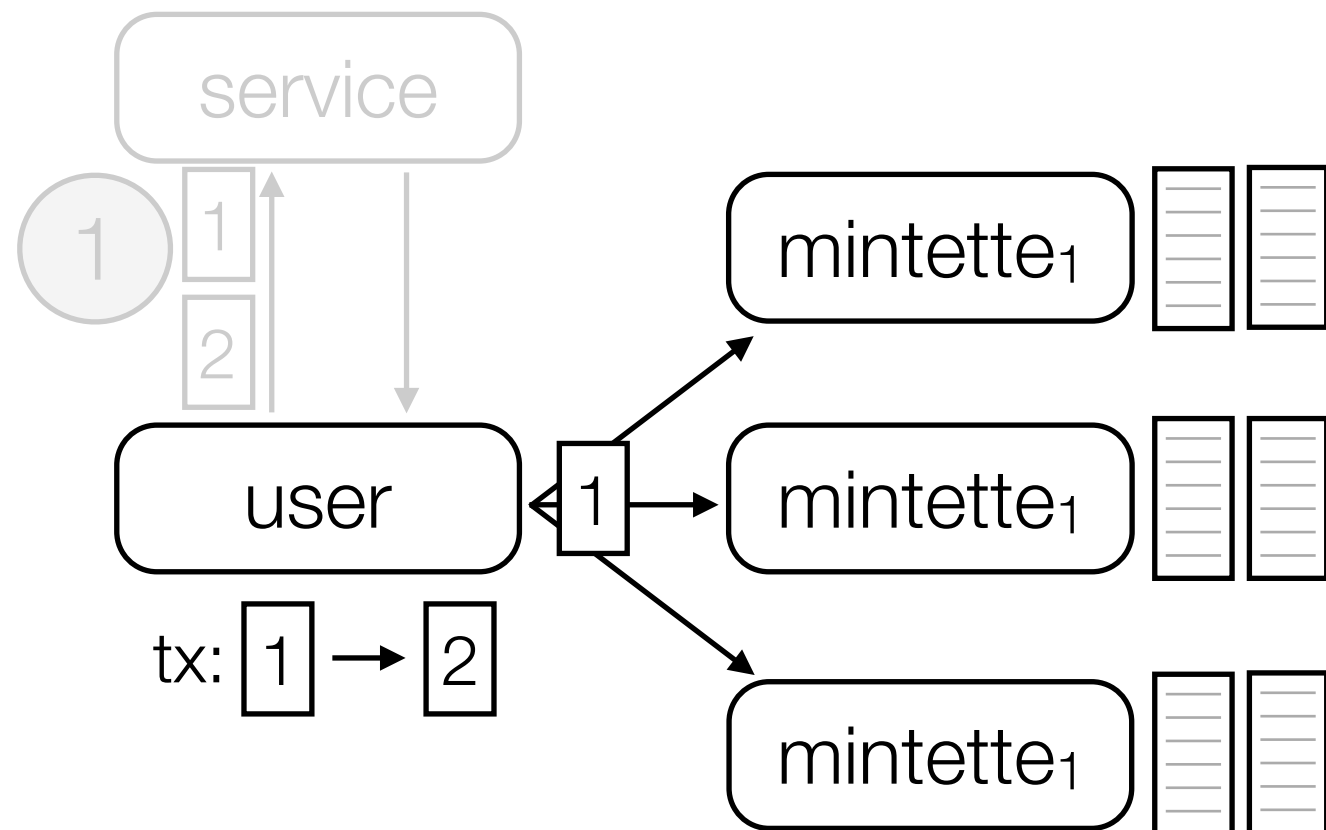
# RSCoin consensus



# simple adaptation of Two-Phase Commit (2PC)

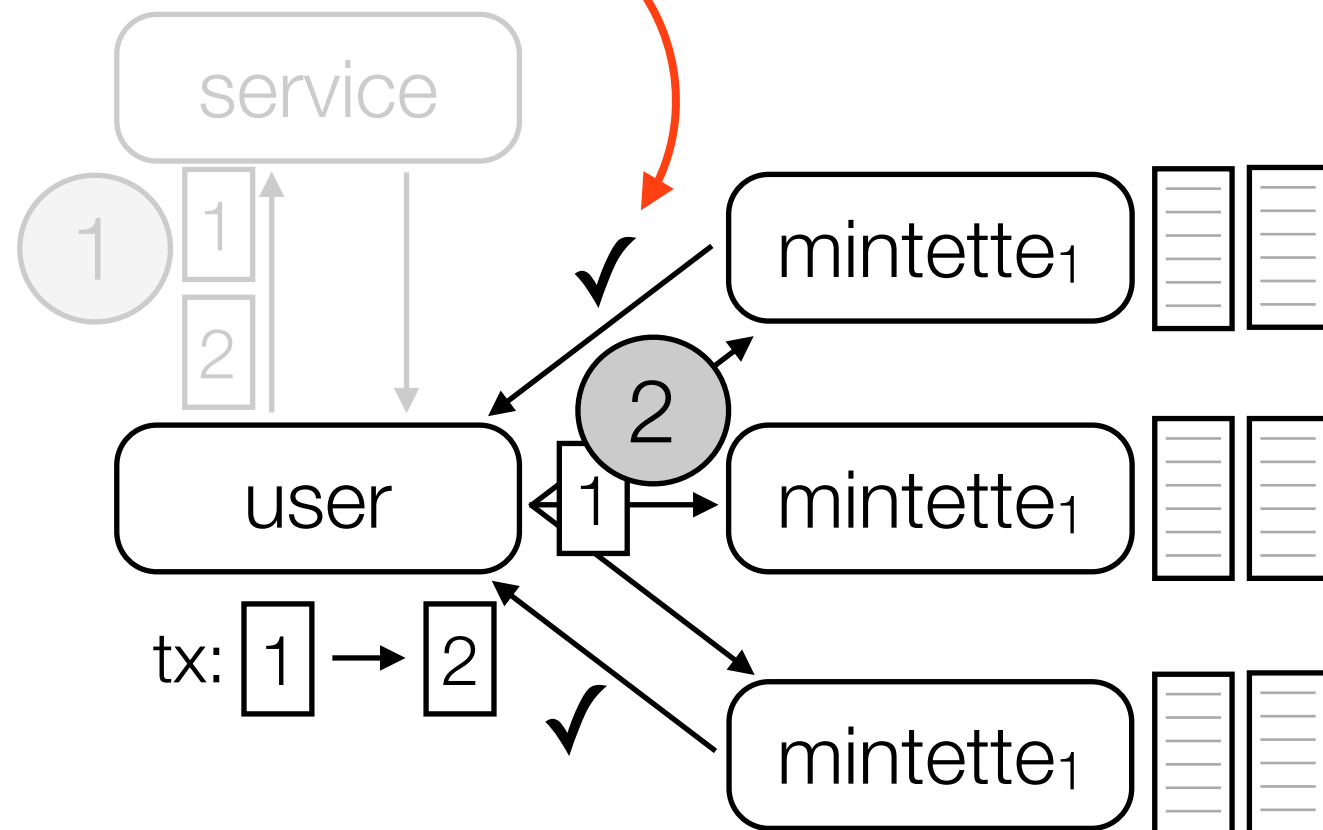


mintettes check for double spending...



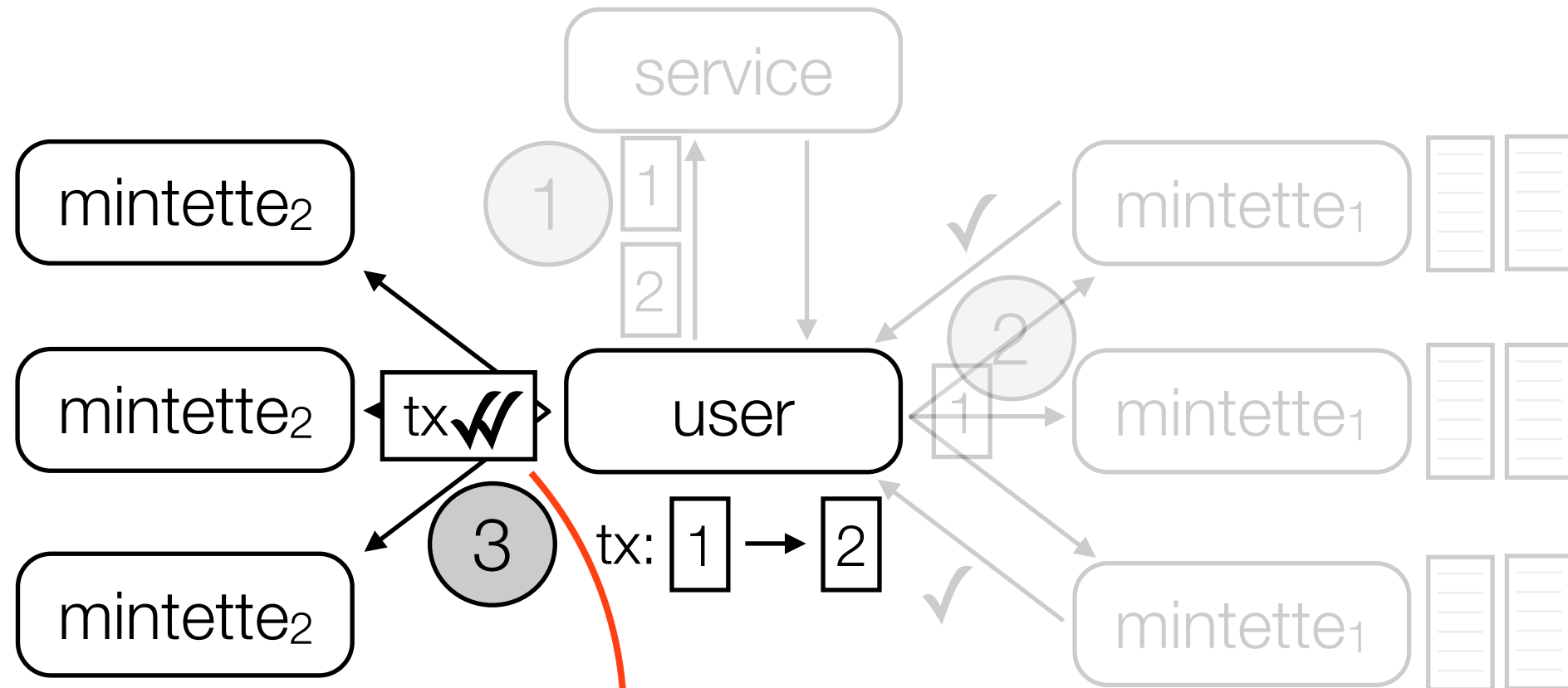
...using lists of unspent transaction outputs (utxo)

signed 'yes' vote



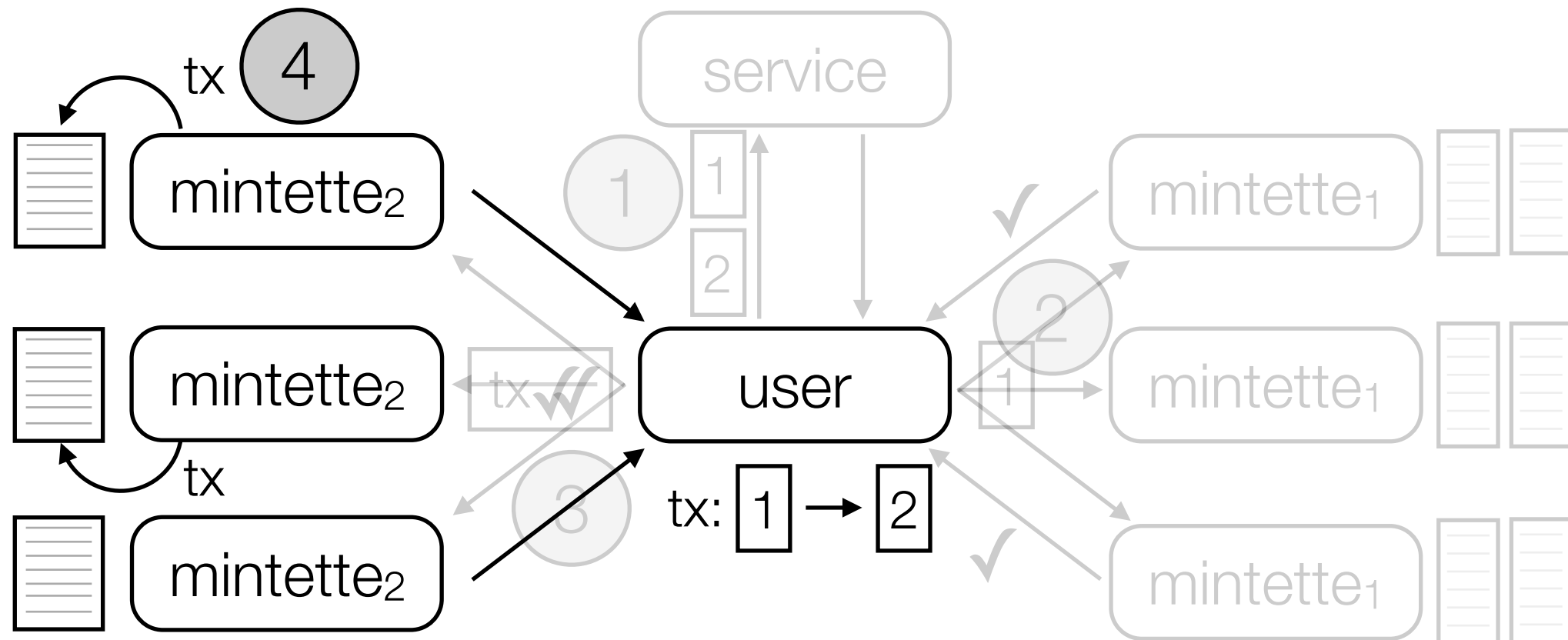


mintettes check validity of bundle by checking for signatures from authorized mintettes...



“bundle of evidence” contains ‘yes’ votes from majority of mintettes in shard

...and if satisfied they add transaction to be **committed** and send back **receipt**



# **security properties**

no double spending (if honest majority per shard)

non-repudiation

auditability (if mintettes log their behavior)

# consensus features

conceptually simple

no broadcast

mintettes communicate only with users

no expensive hashing!

scalable

↑ computational power  $\Rightarrow$  ↑ throughput

# consensus features

$T$  = set of txs generated per second

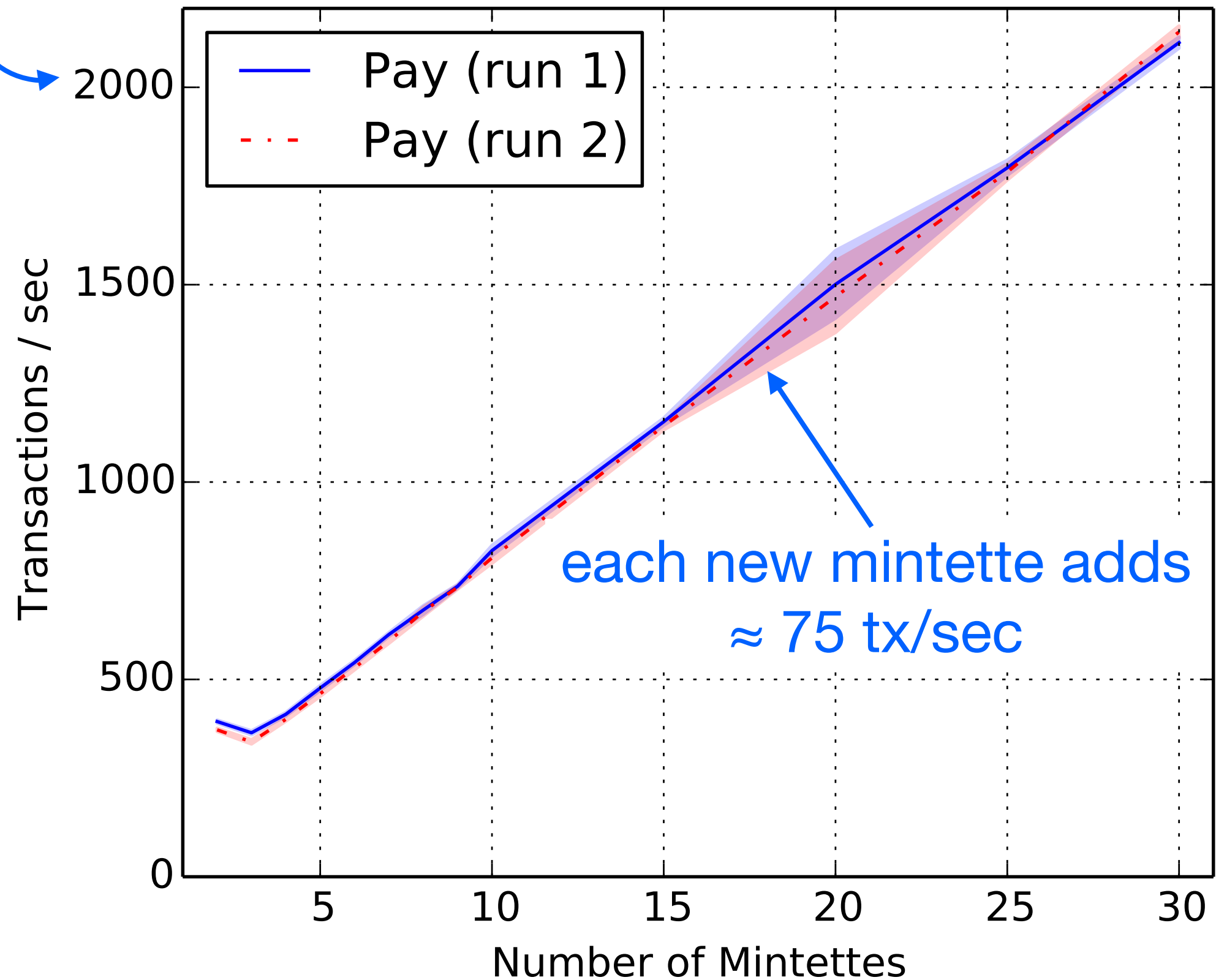
$Q$  = # mintettes per shard

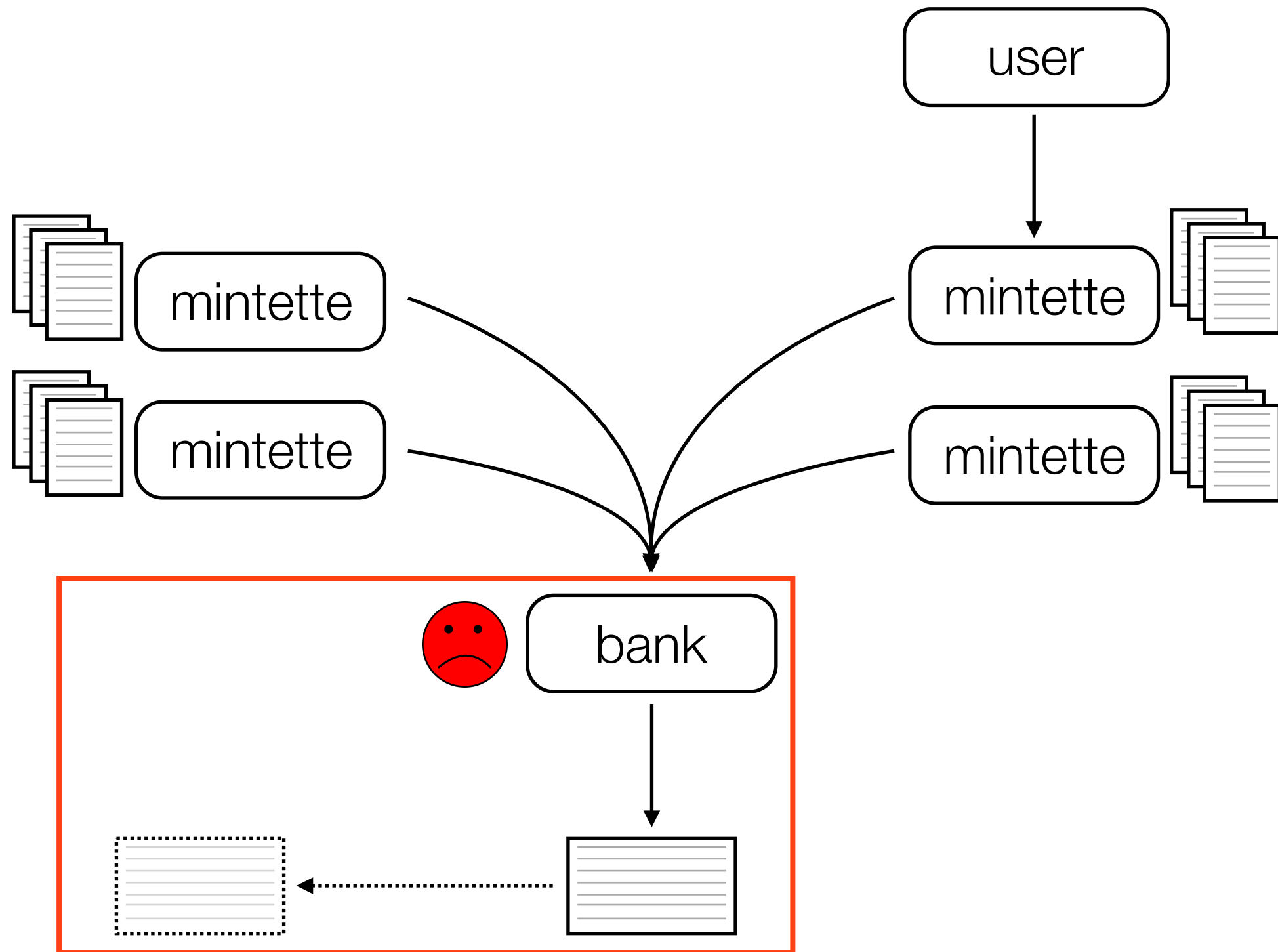
$M$  = # mintettes

$$\text{comm. per mintette per sec} = \frac{\sum_{tx \in T} 2(m_{tx} + 1)Q}{M}$$

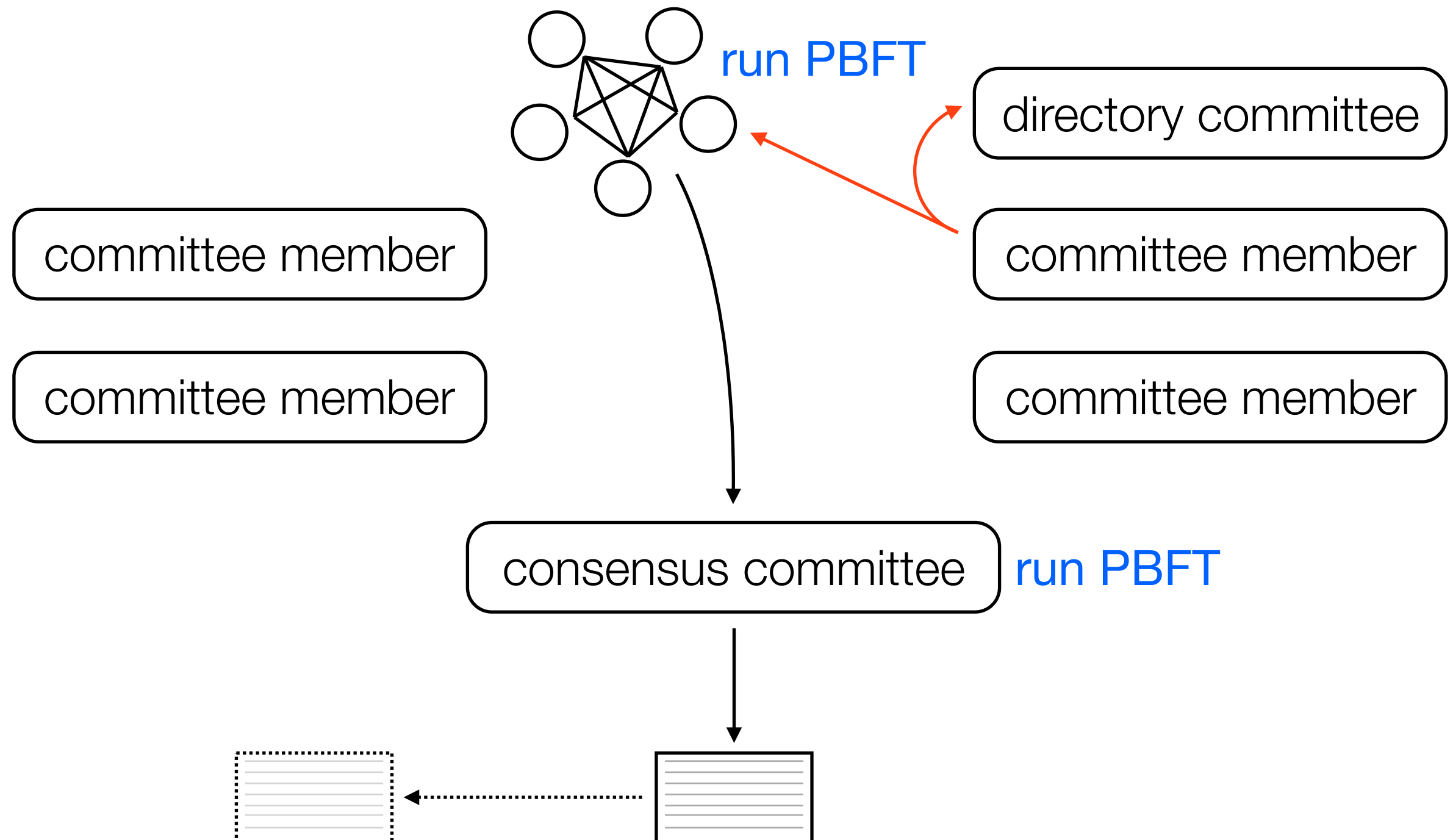
scales infinitely as more mintettes are added!

compared to Bitcoin's 7



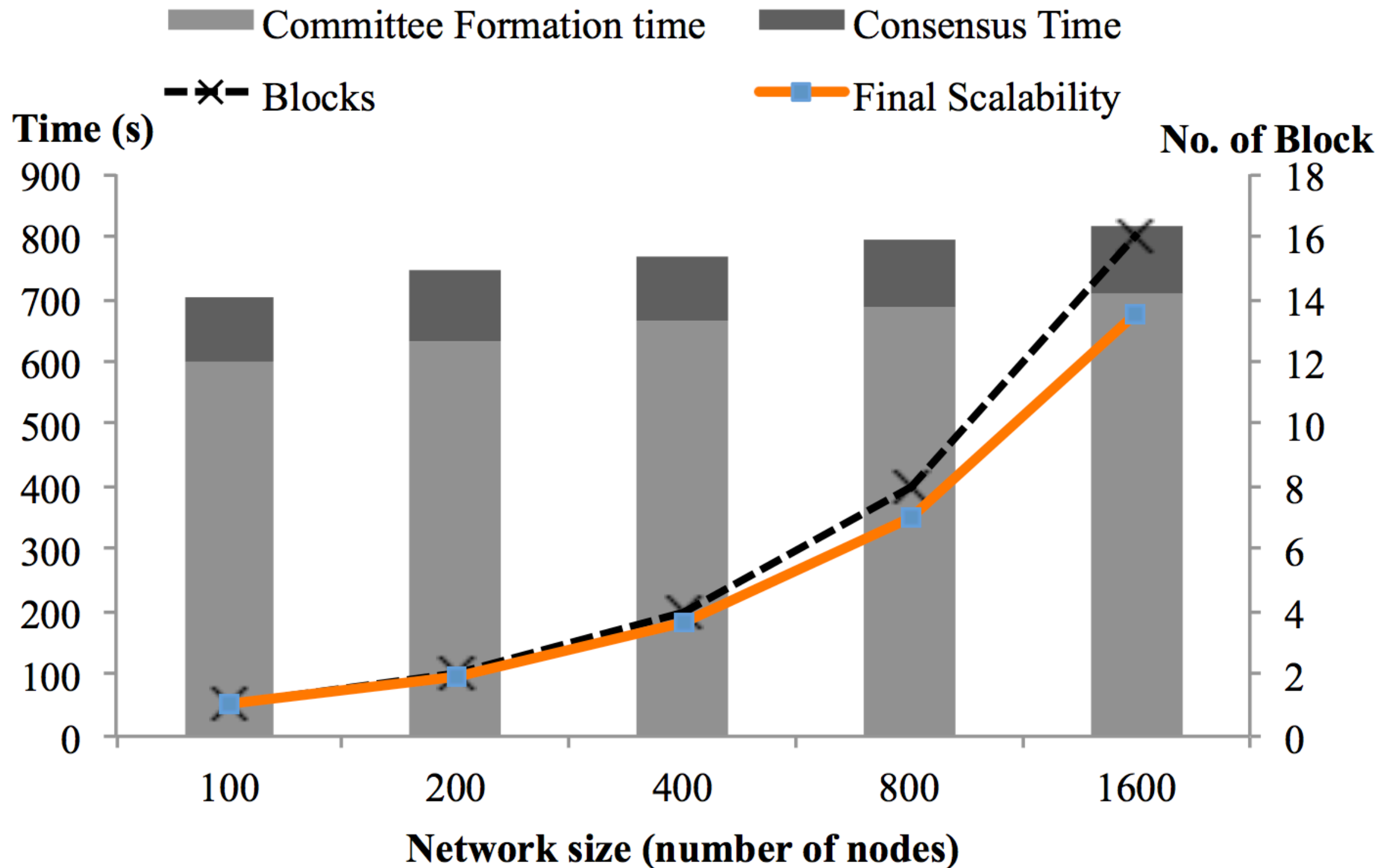


# Elastico [LNZBGS CCS'16]





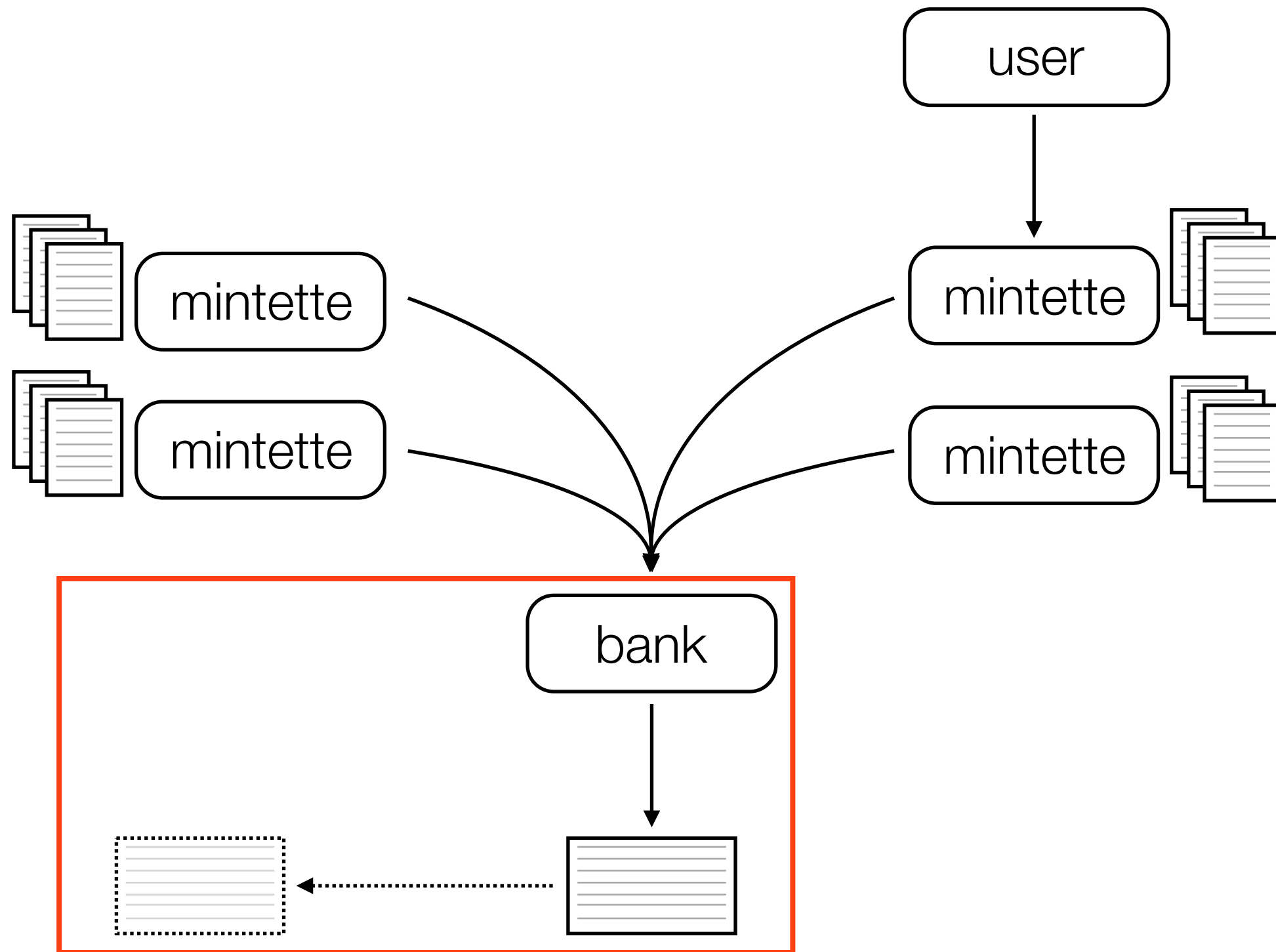
# Elastico [LNZBGS CCS'16]

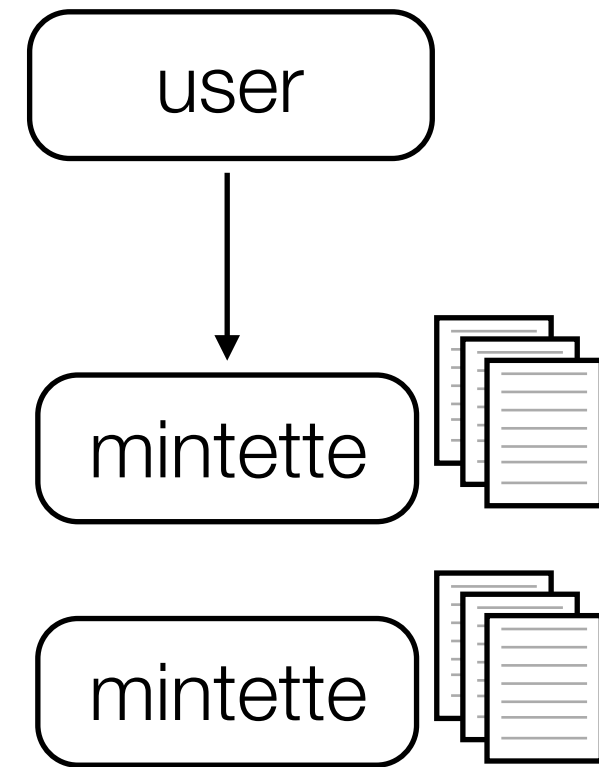
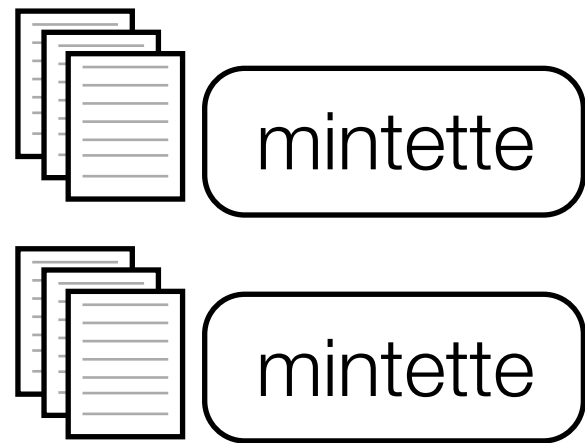


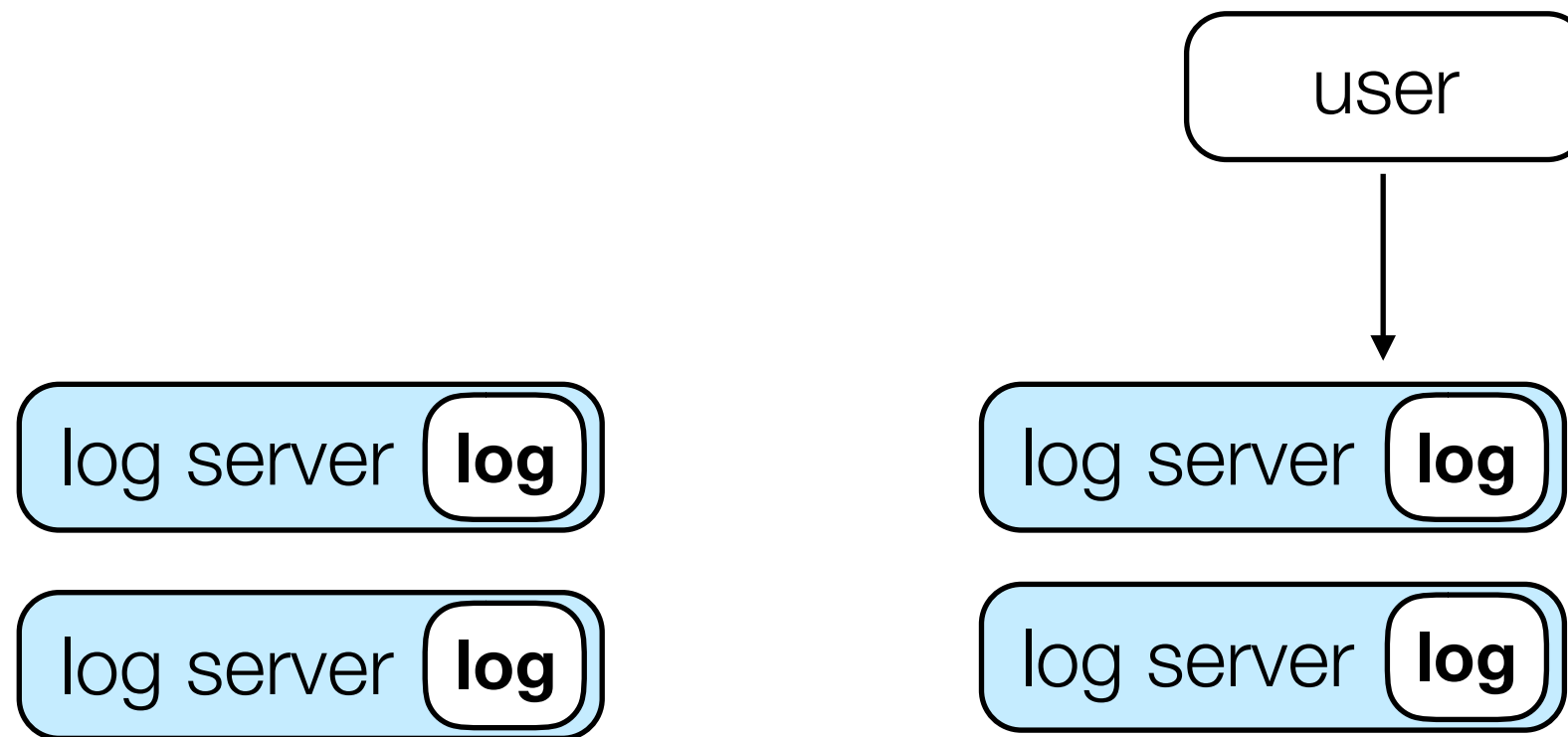
- 10 usability**
- 9 governance**
- 8 comparisons**
- 7 key management**
- 6 agility**
- 5 interoperability**
- 4 scalability**
- 3 cost-effectiveness**
- 2 privacy**
- 1 scalability**

- 10 usability**
- 9 governance**
- 8 comparisons**
- 7 key management**
- 6 agility**
- 5 interoperability**
- 4 scalability**
- 3 cost-effectiveness**
- 2 privacy**
- 1 scalability**

# RSCoin [DM NDSS'16]



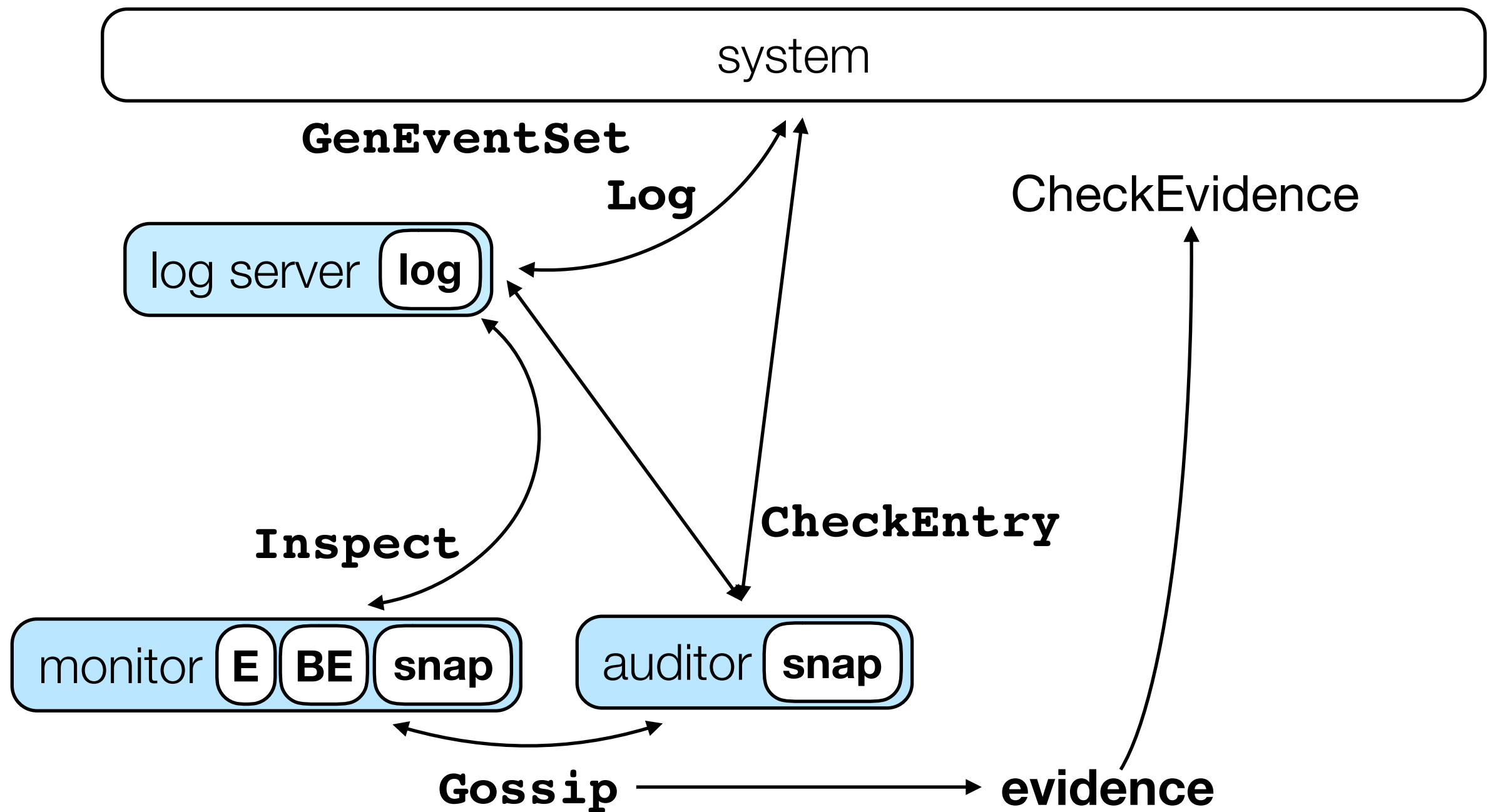


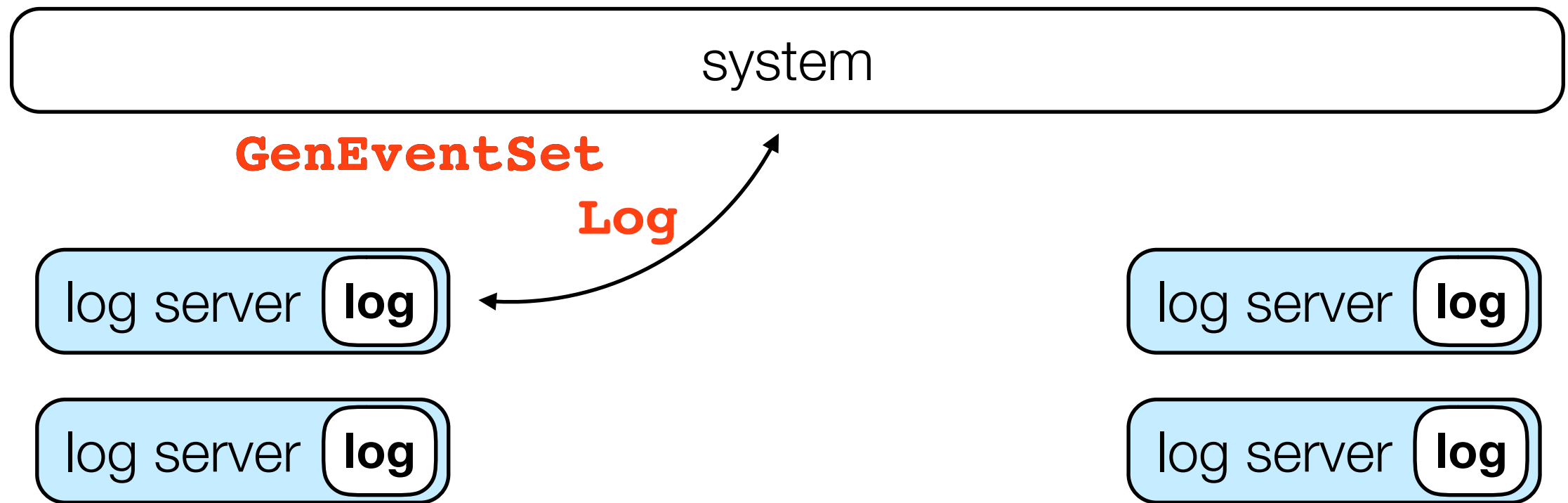


no unified log  $\Rightarrow$  no need for consensus

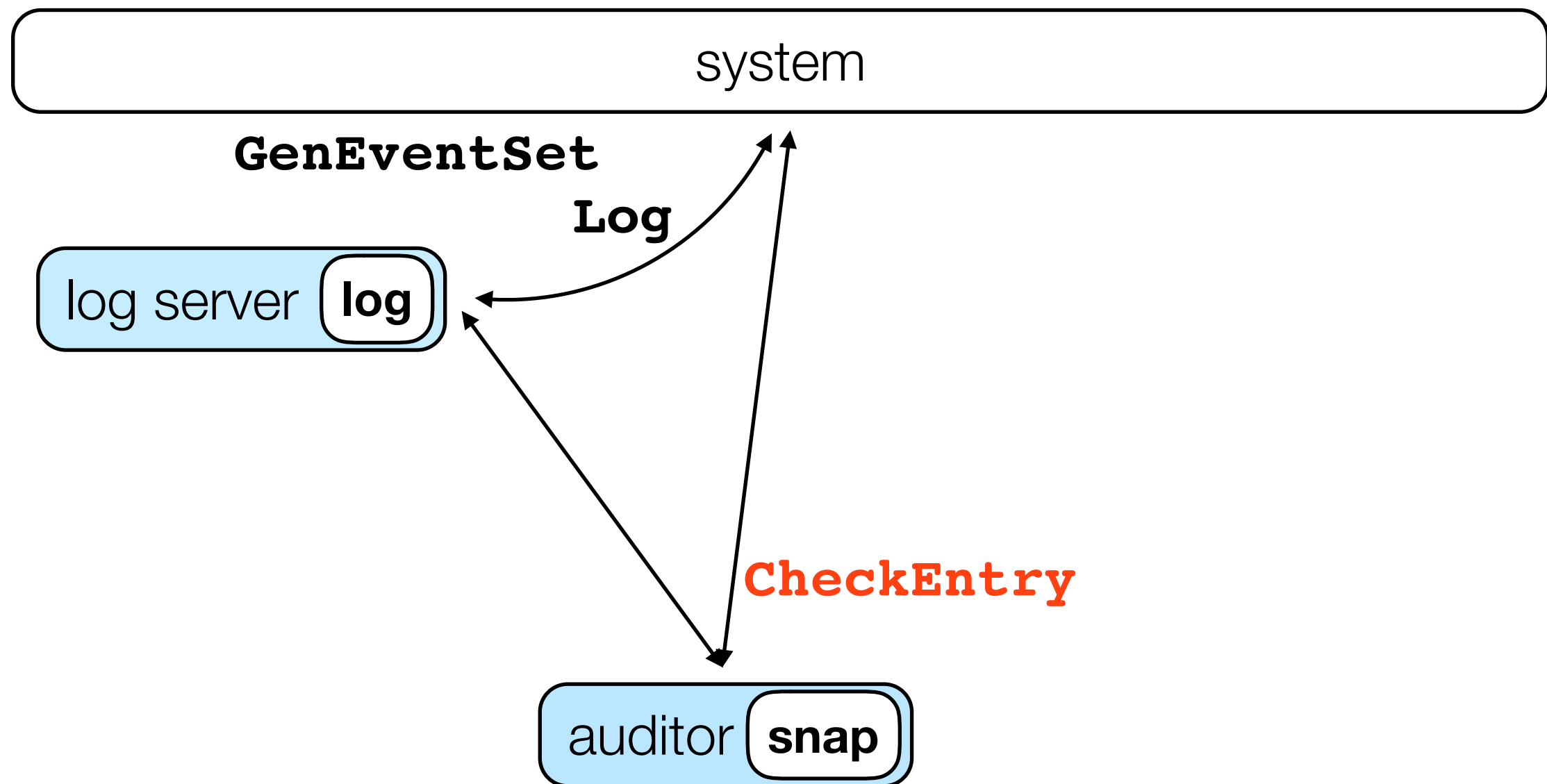
can (retroactively) detect inconsistencies between logs

# transparency overlays [CM CCS'16]



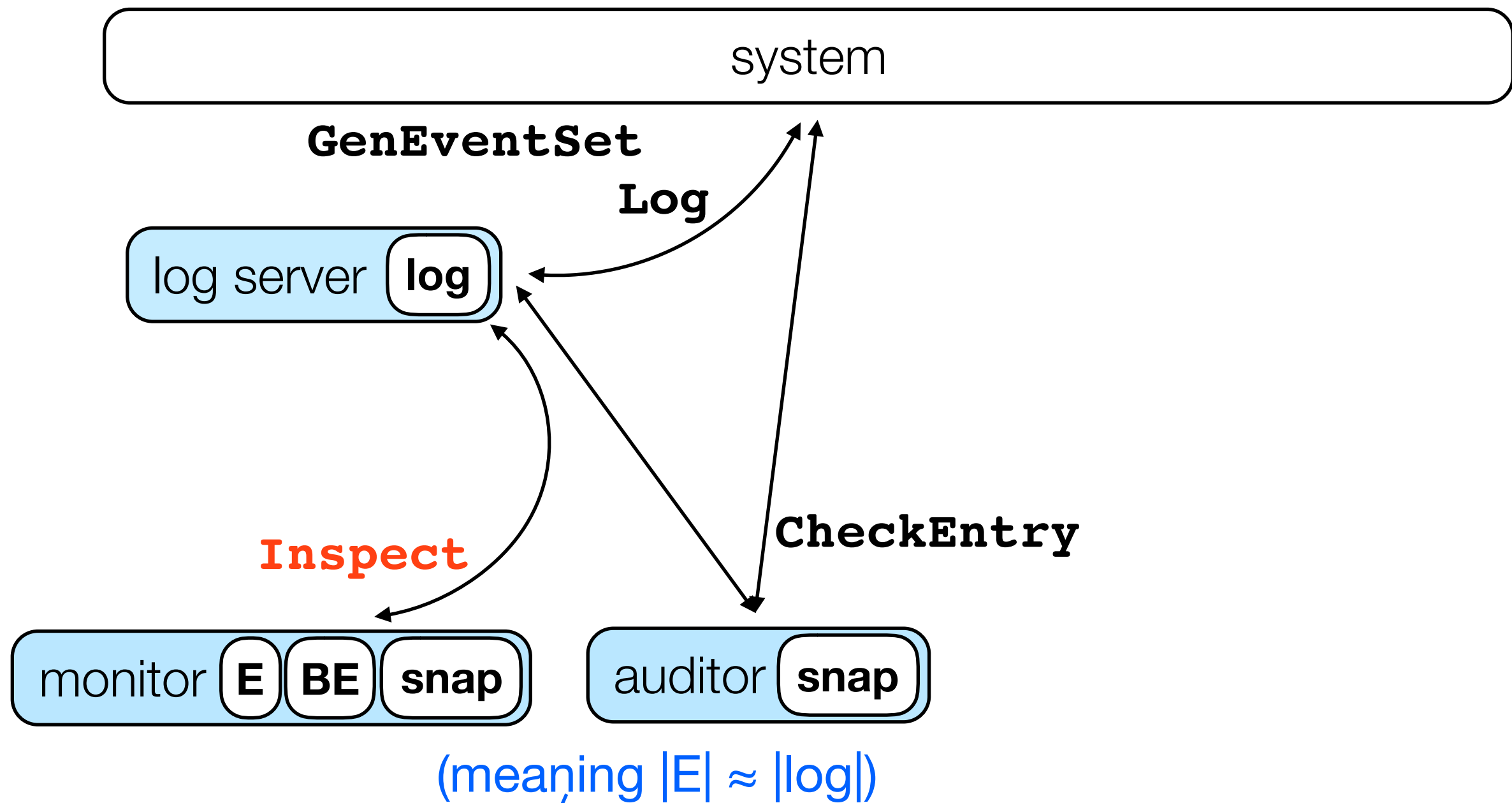




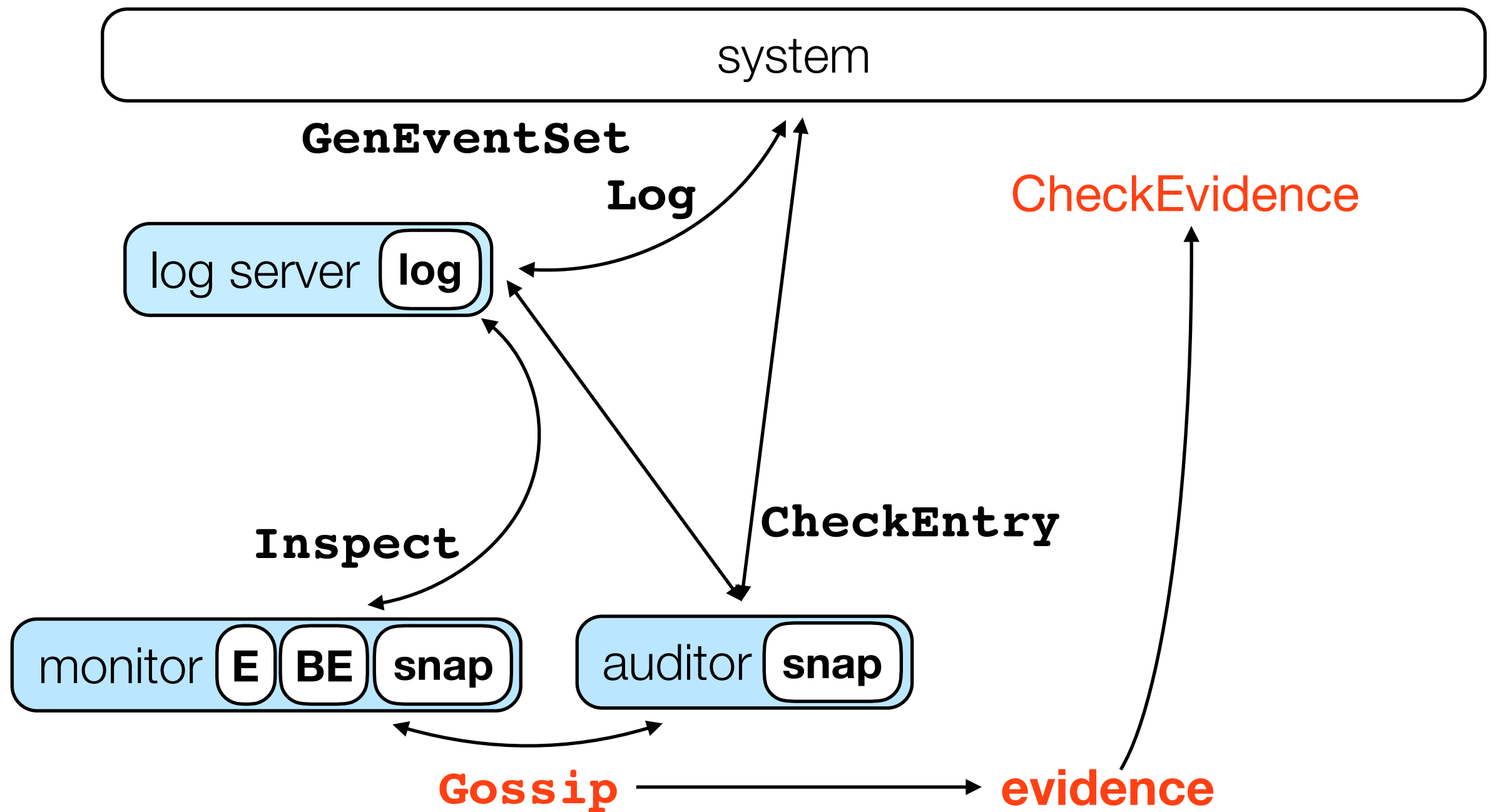


(meaning  $|\text{snap}| \ll |\text{log}|$ )

auditors (efficiently) determine if events are in the log



monitors (inefficiently) detect bad events in the log



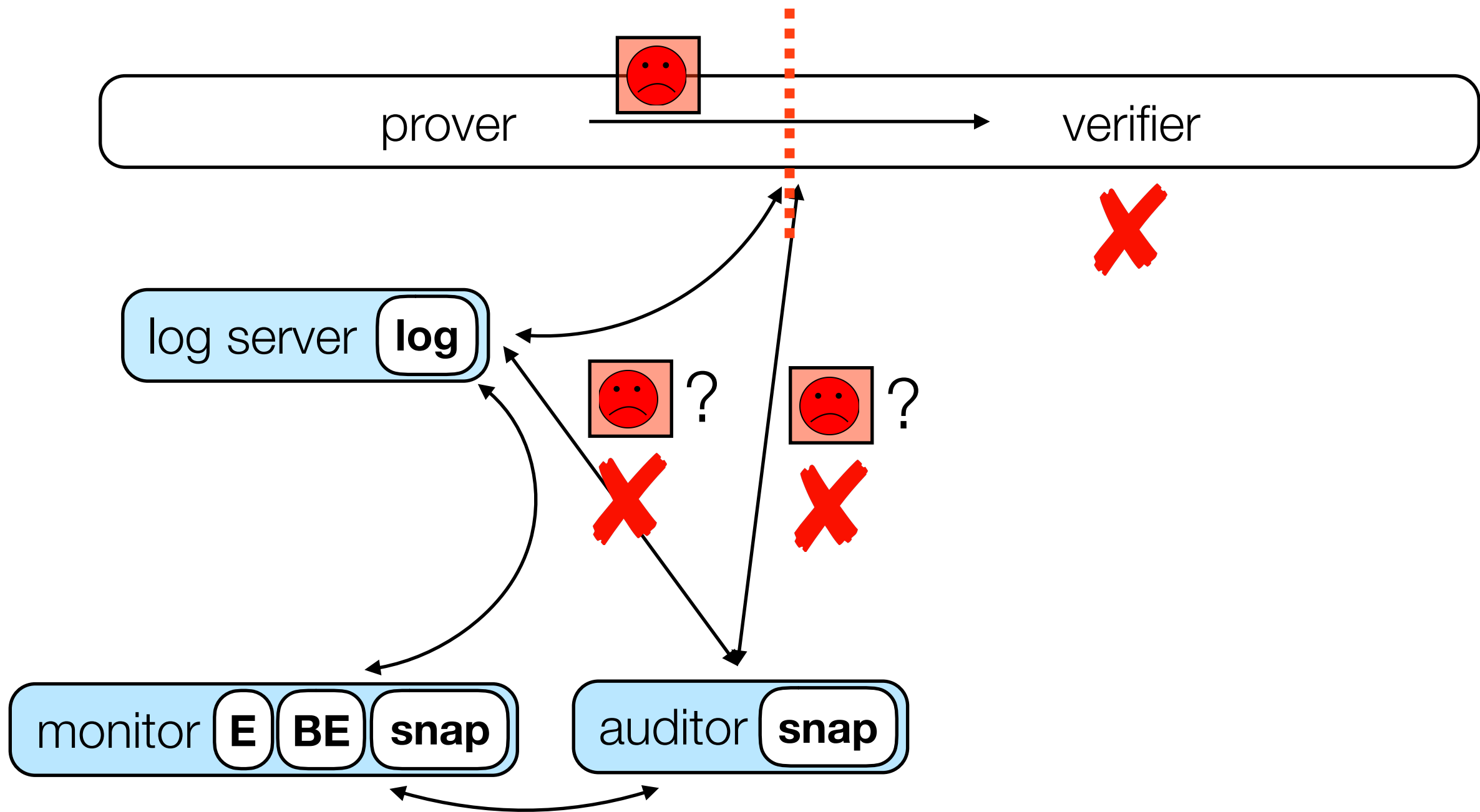
auditors and monitors ensure consistent view of log  
(can output evidence of inconsistencies)

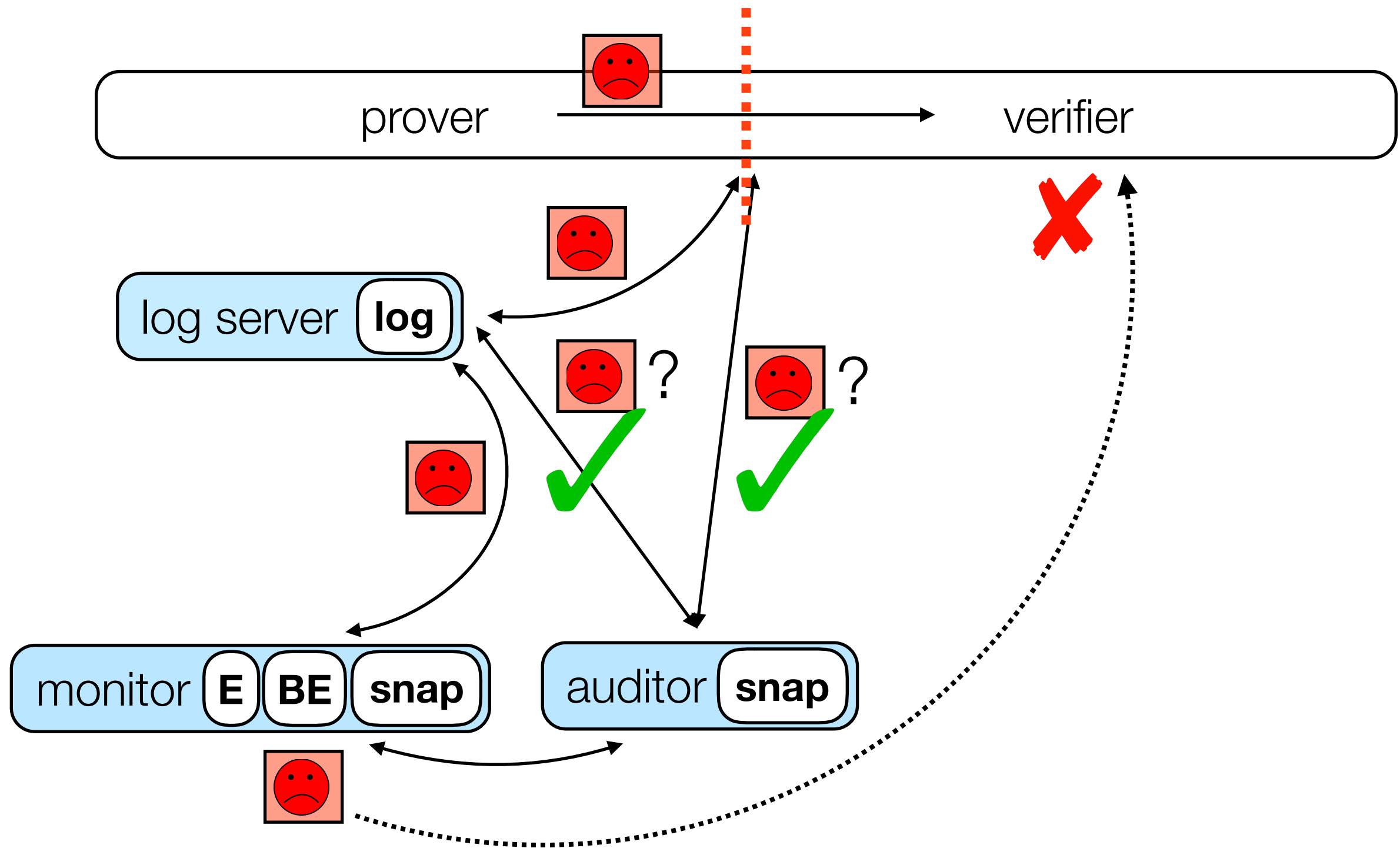
# **security properties**

consistency: log server can't offer different views of log

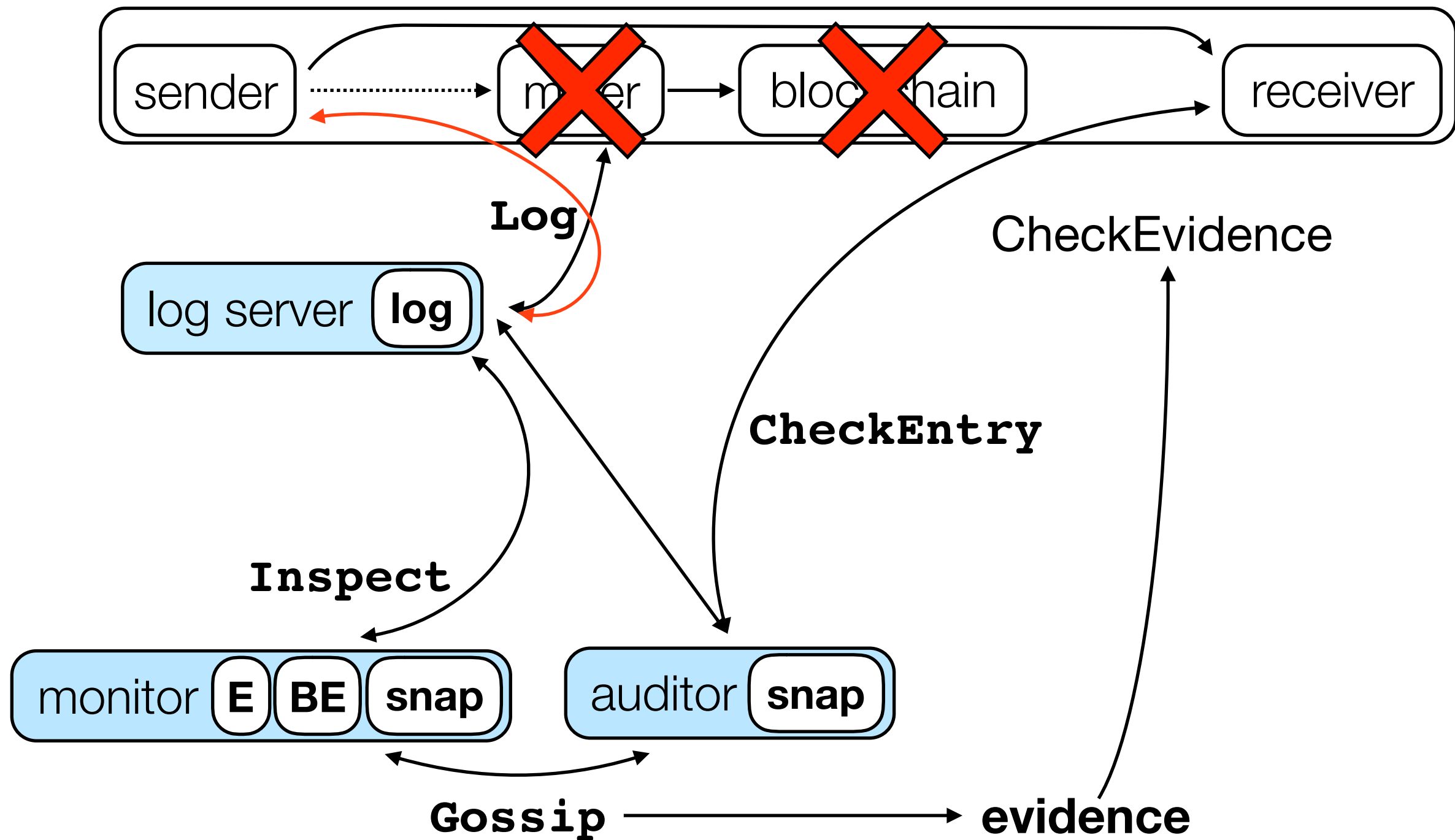
non-frameability: auditor and monitor can't frame the log

accountability: log server is held to its promises



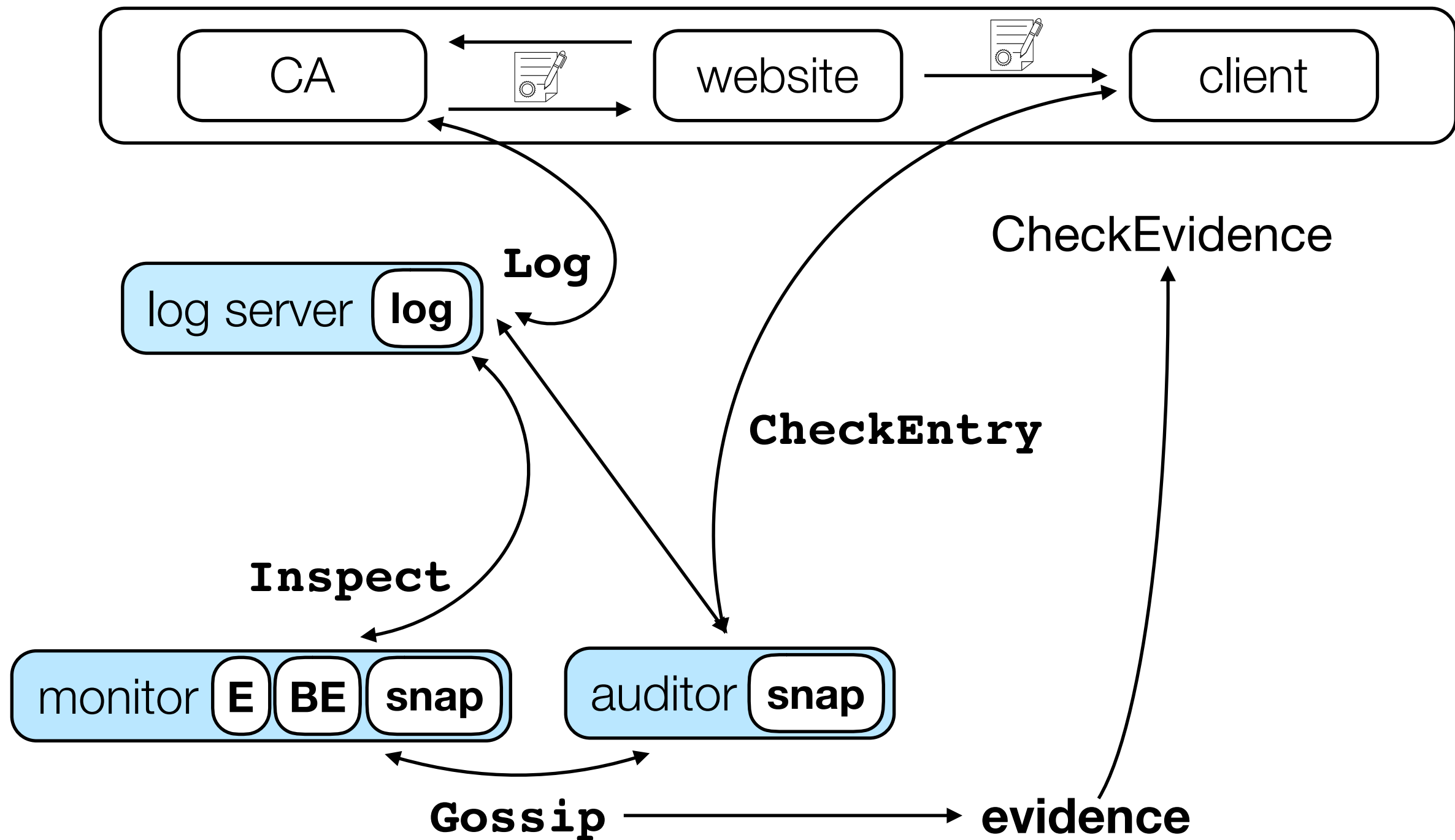


# Bitcoin



sender and receiver don't need to store blockchain  
gives rise to hybrid system ( $\approx$ RS Coin) with no mining

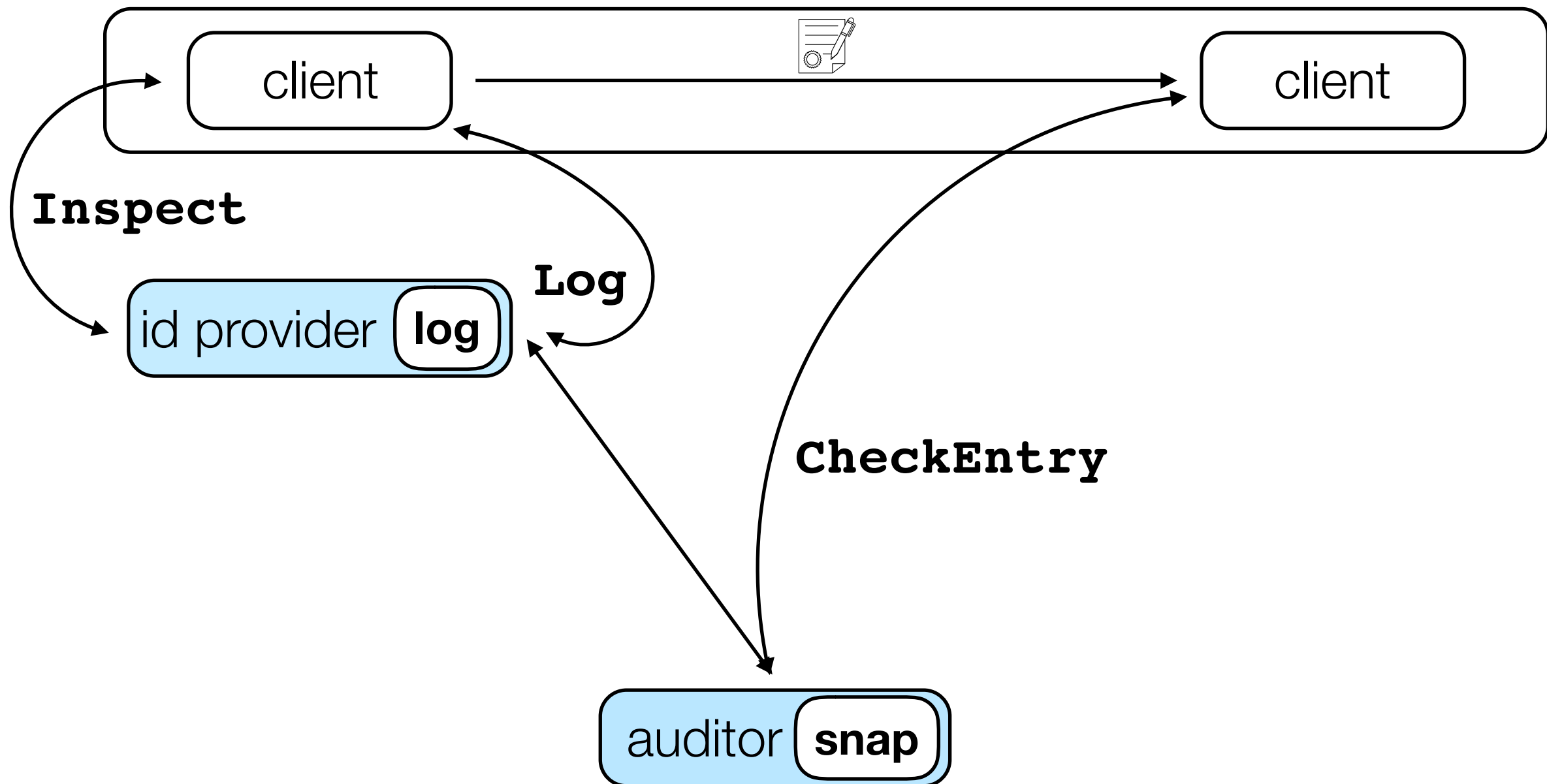
# Certificate Transparency [LL13]



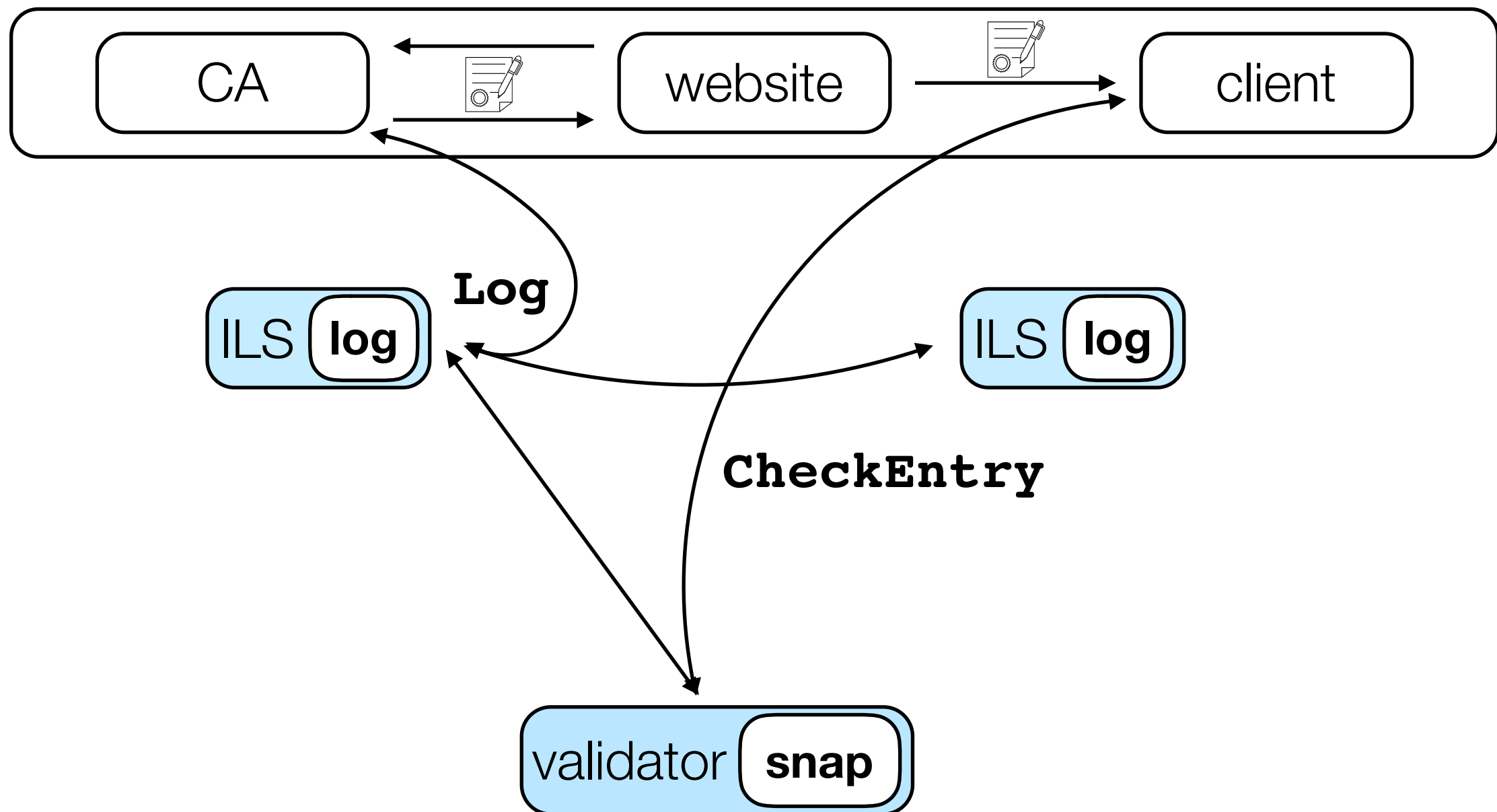
bad certificate issuance is exposed  
⇒ clients are less likely to accept bad certificates

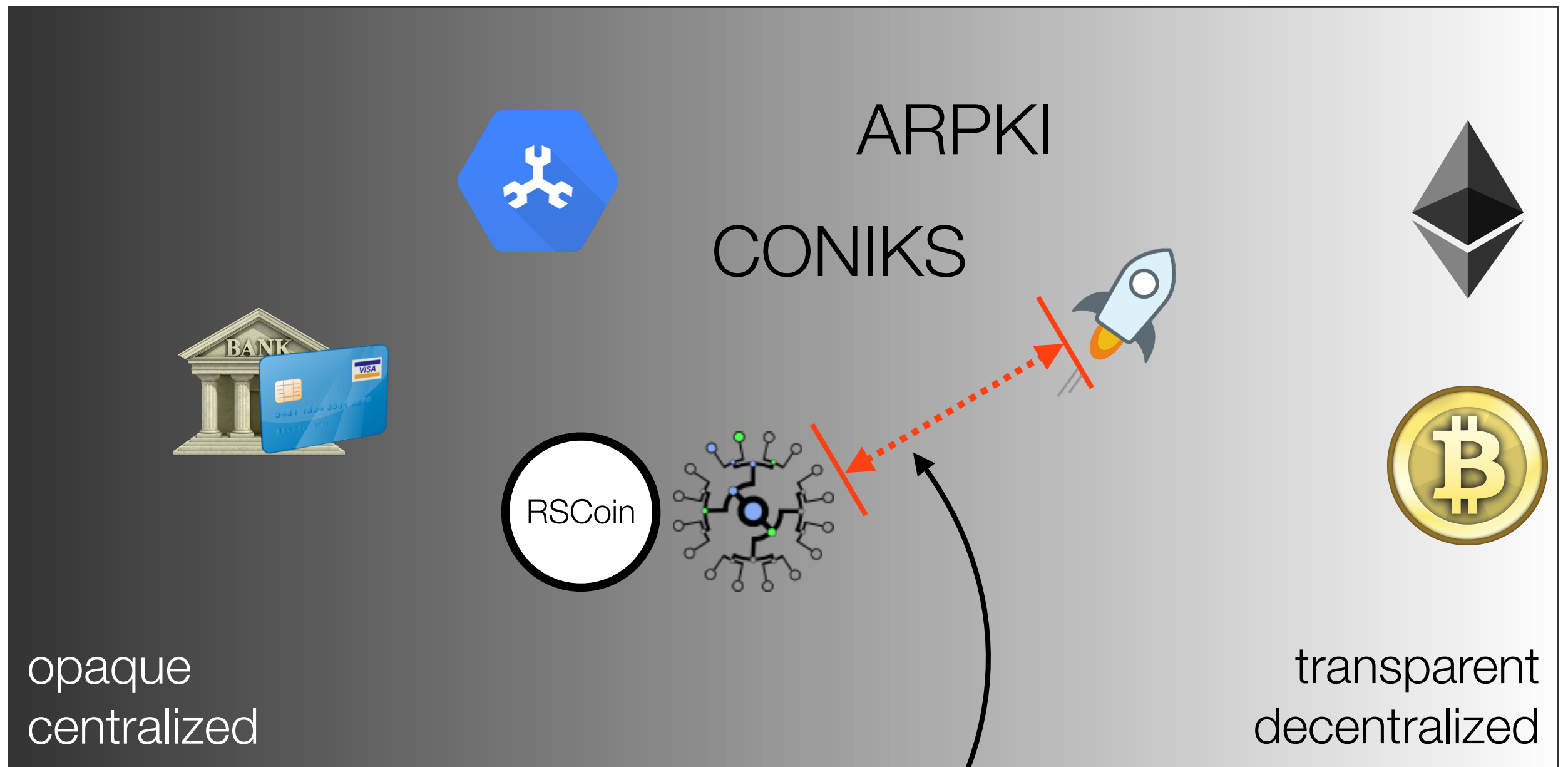


# CONIKS [MBBFF USENIX Sec'16]



# ARPKI [BCKPSS CCS'13]





# security properties

(transparency overlays)

consistency



non-frameability



accountability



privacy (of what)?

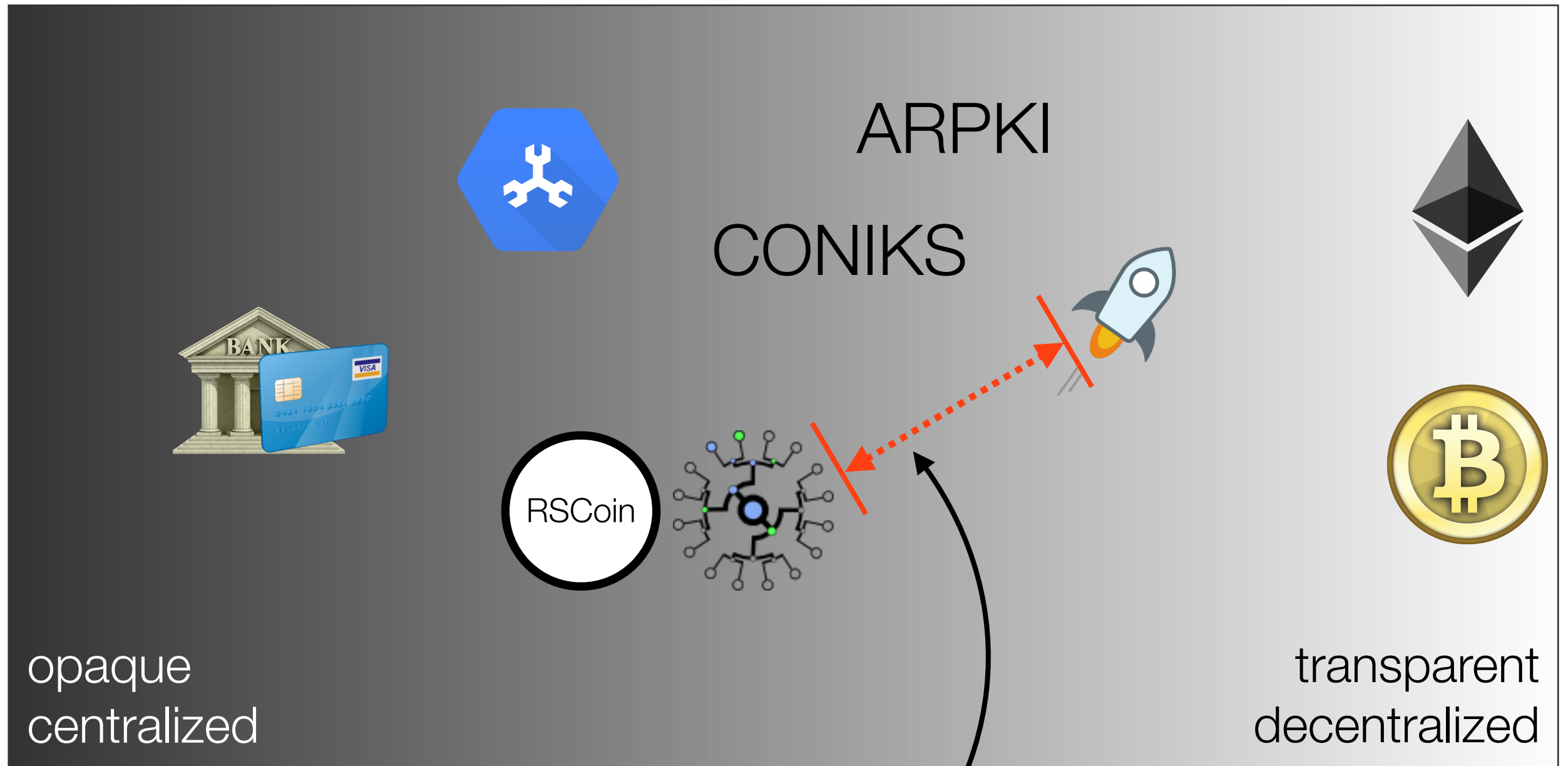
(RSCoin)

no double spending

non-repudiation

auditability

privacy (of what)?



what security properties to look for?

**10 usability**

**9 governance**

**8 comparisons**

**7 key management**

**6 agility**

**5 interoperability**

**4 scalability**

**3 cost-effectiveness**

**2 privacy**

**1 scalability**

- 10 usability**
- 9 governance**
- 8 comparisons**
- 7 key management**
- 6 agility**
- 5 interoperability**
- 4 scalability**
- 3 cost-effectiveness**
- 2 privacy**
- 1 scalability**

Thanks! Any questions?