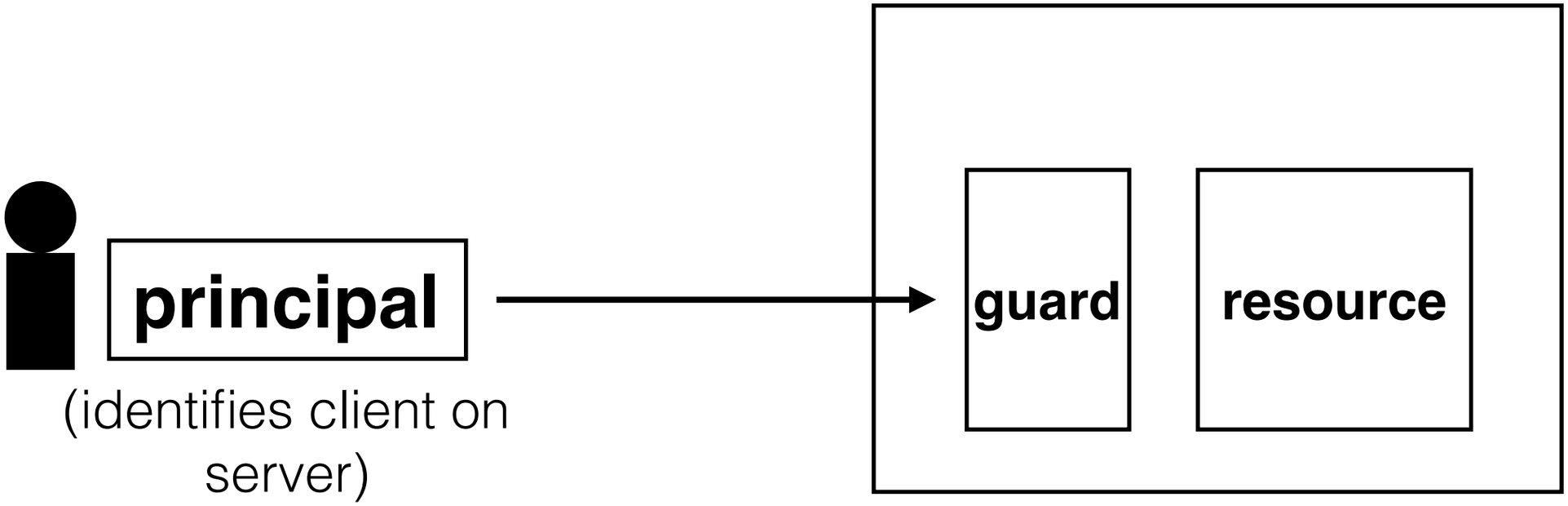
6.033 Spring 2018

Lecture #24

- **Anonymity and Digital Currency**


principal
(identifies client on server)
guard
resource

Bitcoin and Tor

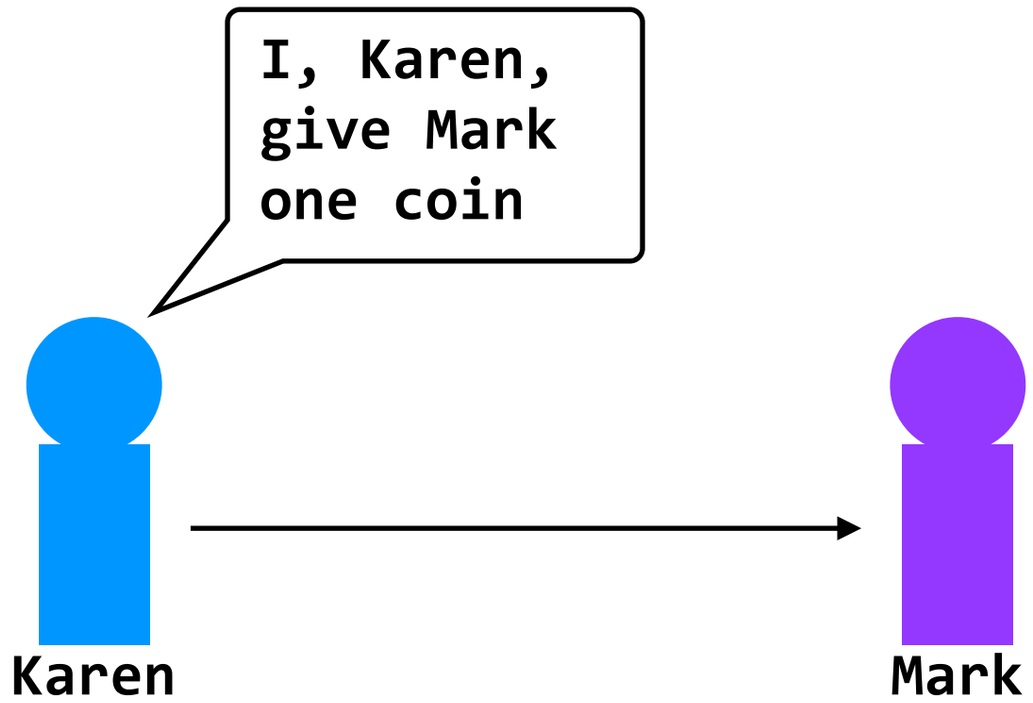
two technologies that deal, either directly or somewhat-tangentially, with **anonymity**

digital currency

currency

decentralized currency

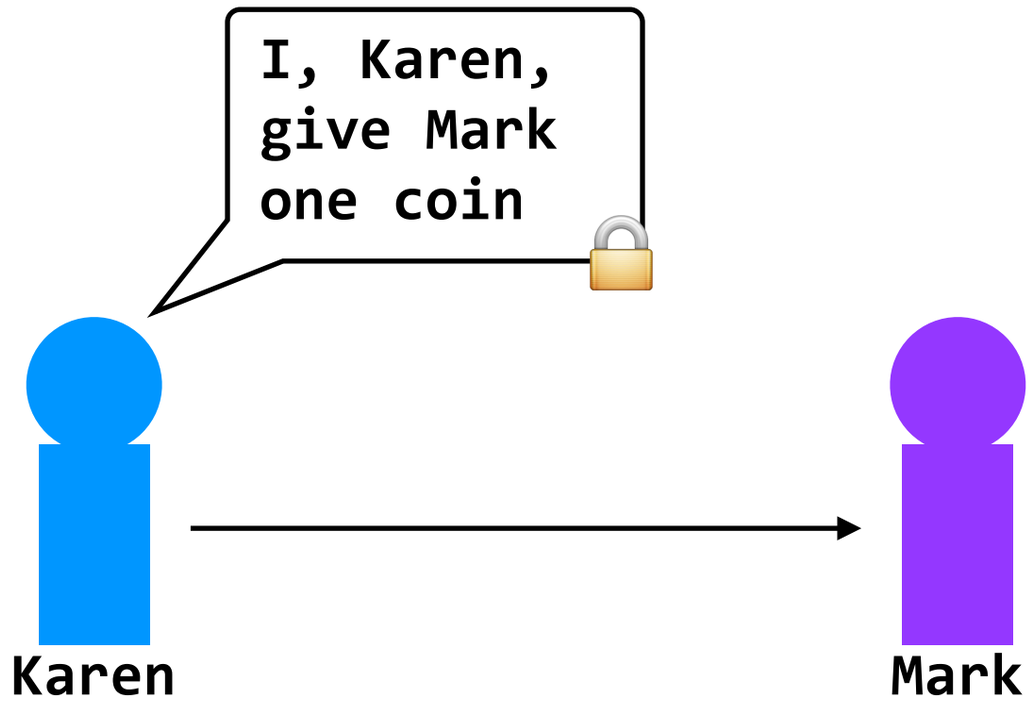
can we avoid having a centralized bank?



problem: easily forgeable

decentralized currency

can we avoid having a centralized bank?



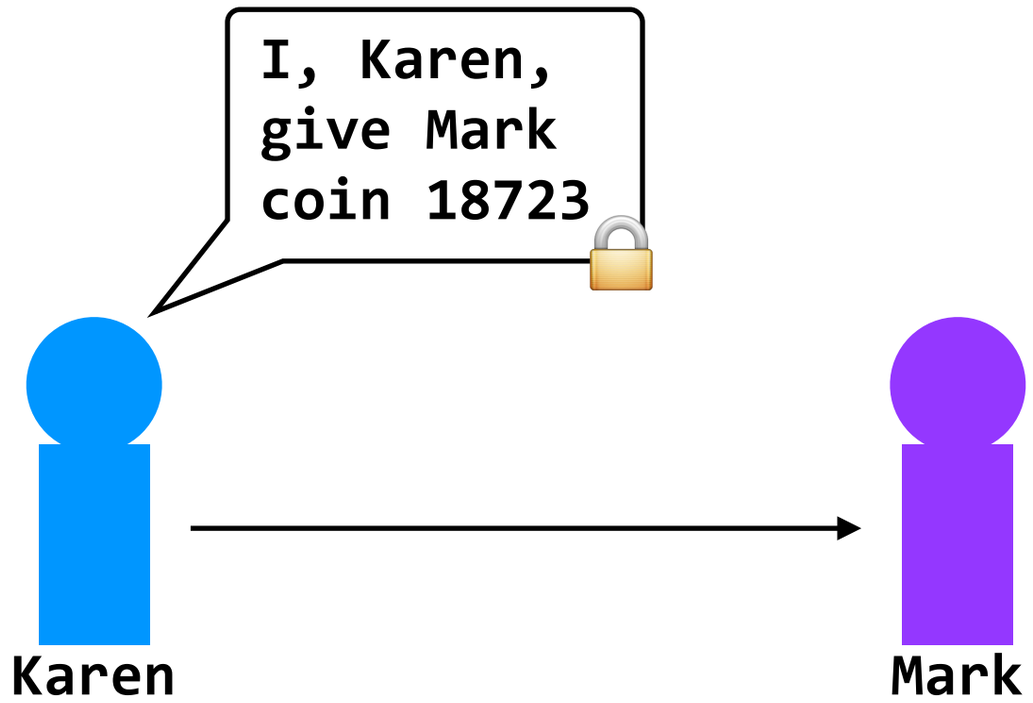
use digital signatures

Karen signs the message with her secret key

problem: replay attacks

decentralized currency

can we avoid having a centralized bank?

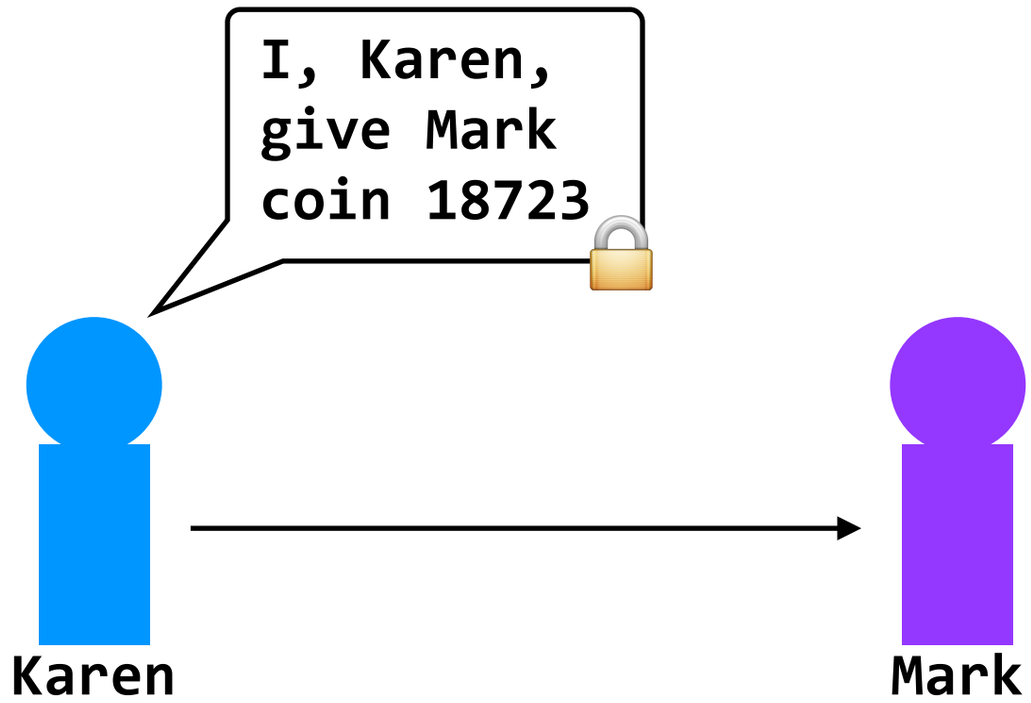


use sequence numbers

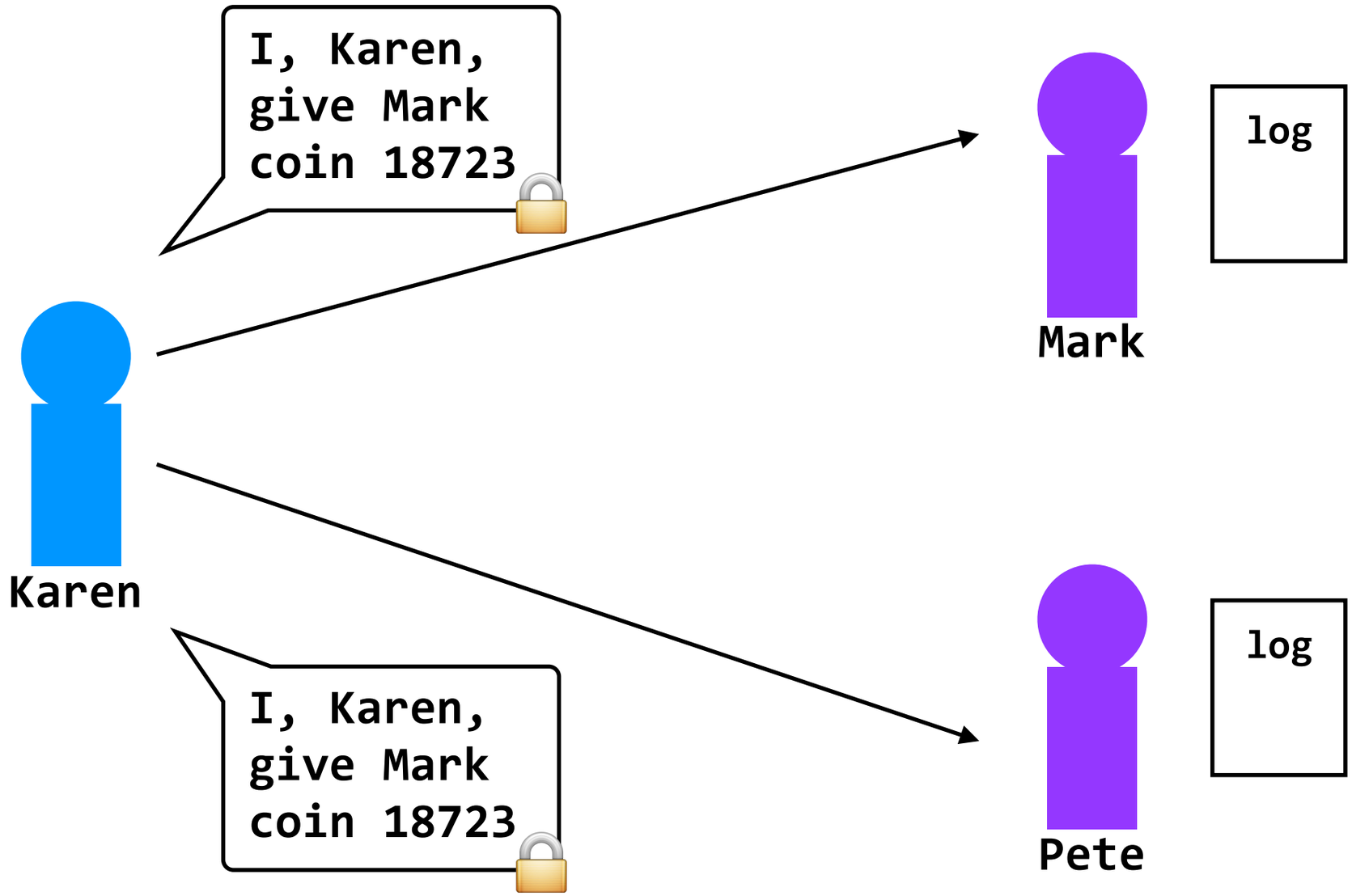
(serial numbers)

decentralized currency

can we avoid having a centralized bank?



problem: how does Mark know that Karen owns coin 18723?

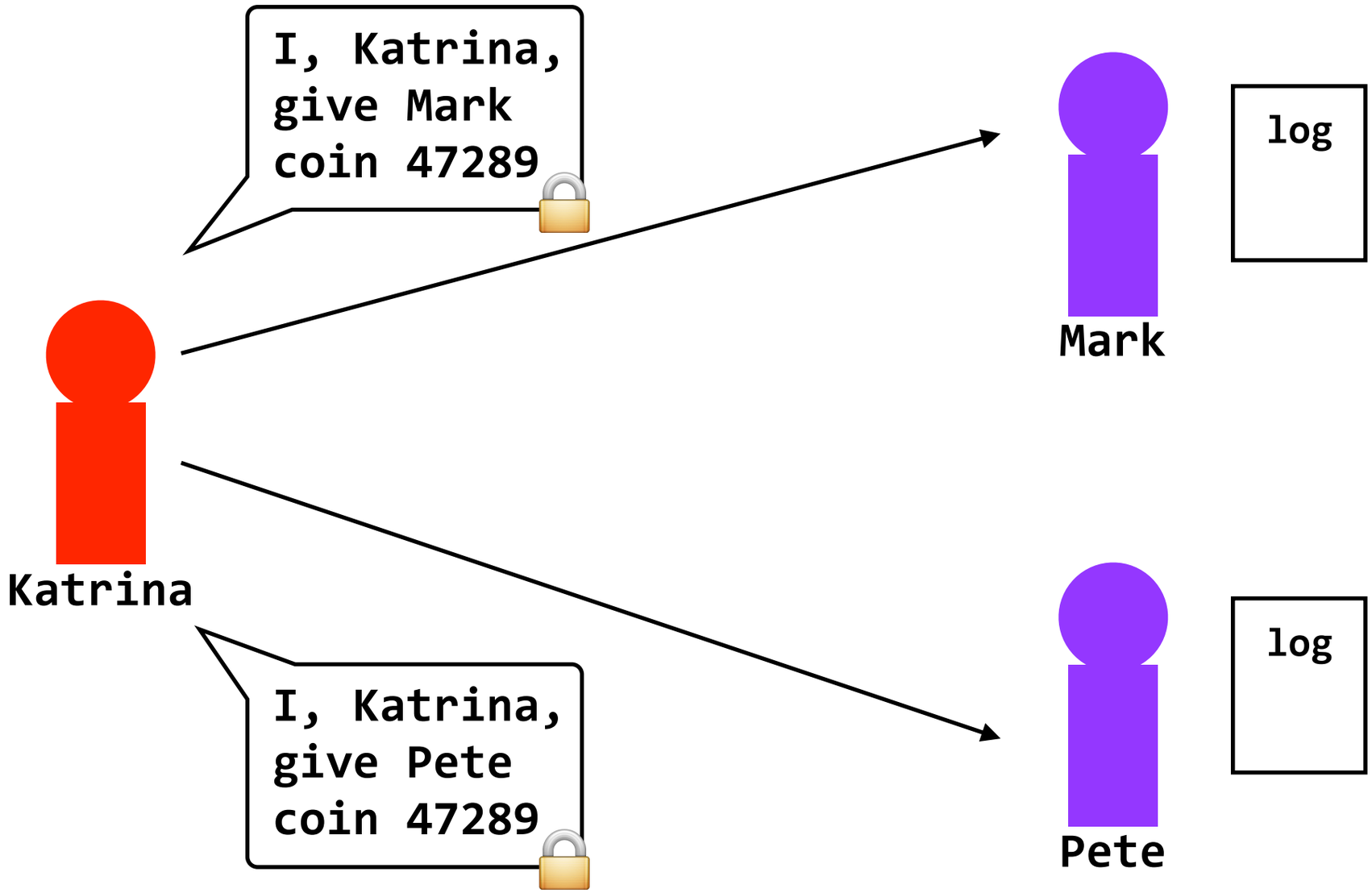


broadcast transactions

everyone in the network keeps a log of all transactions

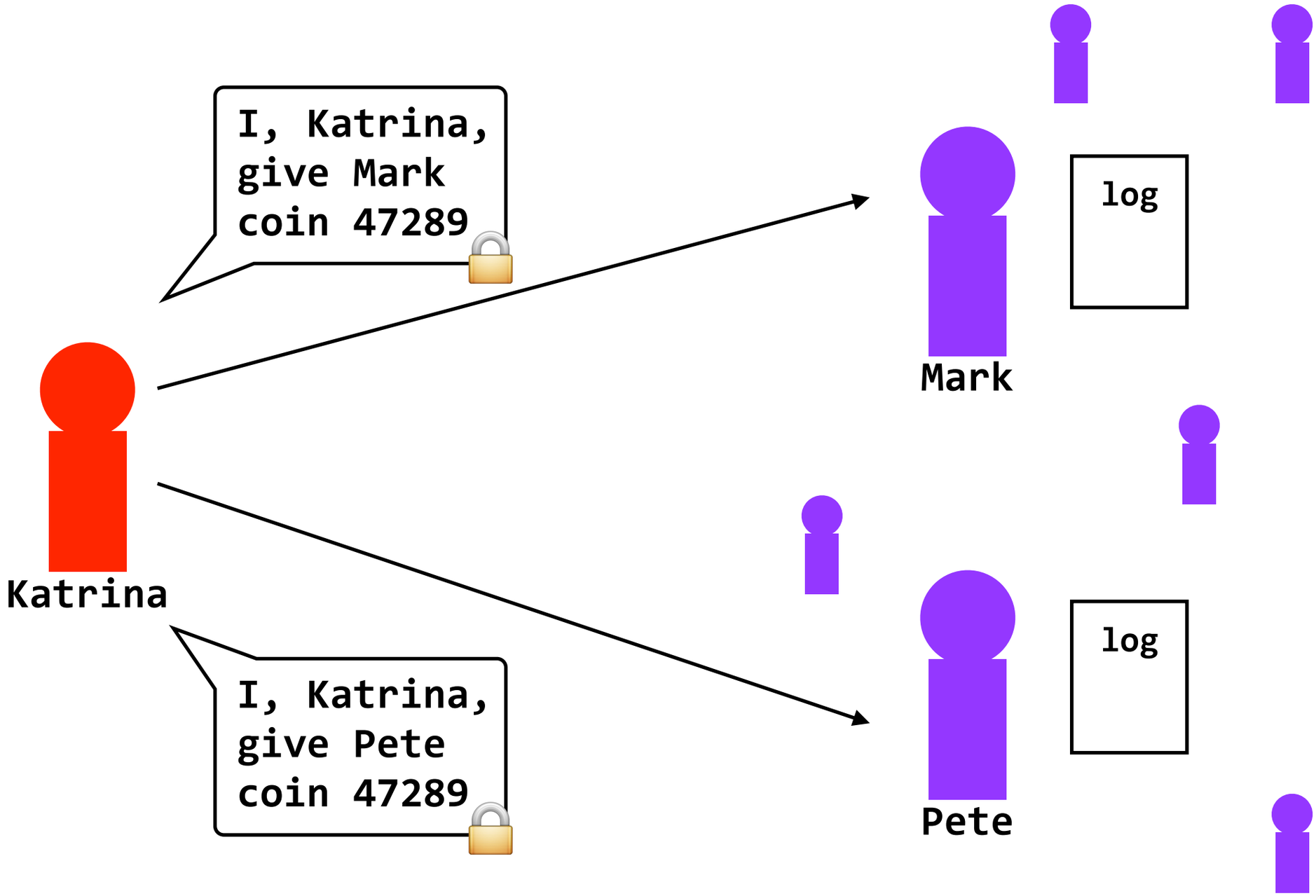
(a “decentralized, public log”)

For ordering, each transaction will contain a hash of the transaction that came before it

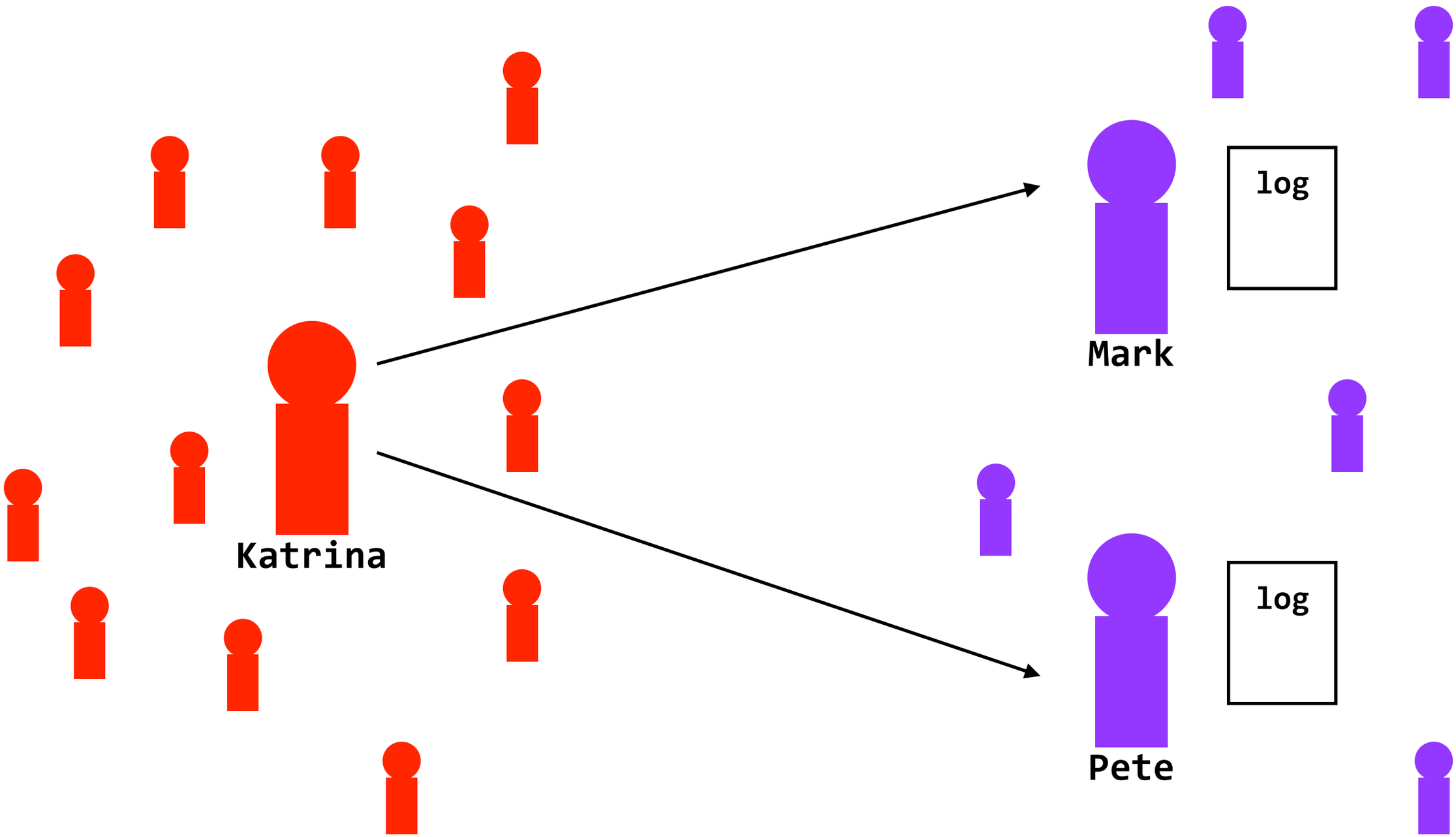


problem: what if Katrina tries to spend with Mark and Pete at the same time?

(before either party has a chance to publish the transaction)

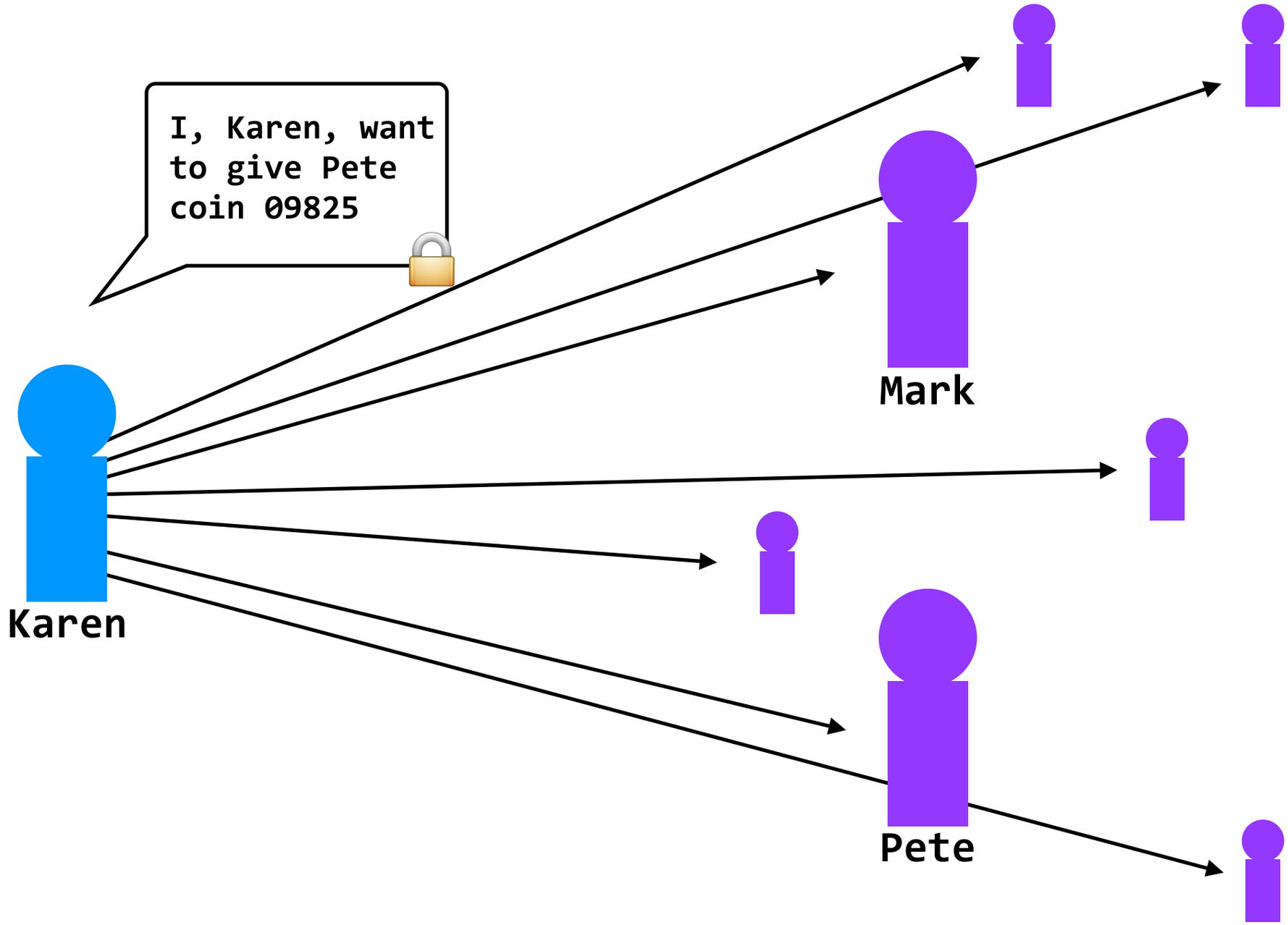


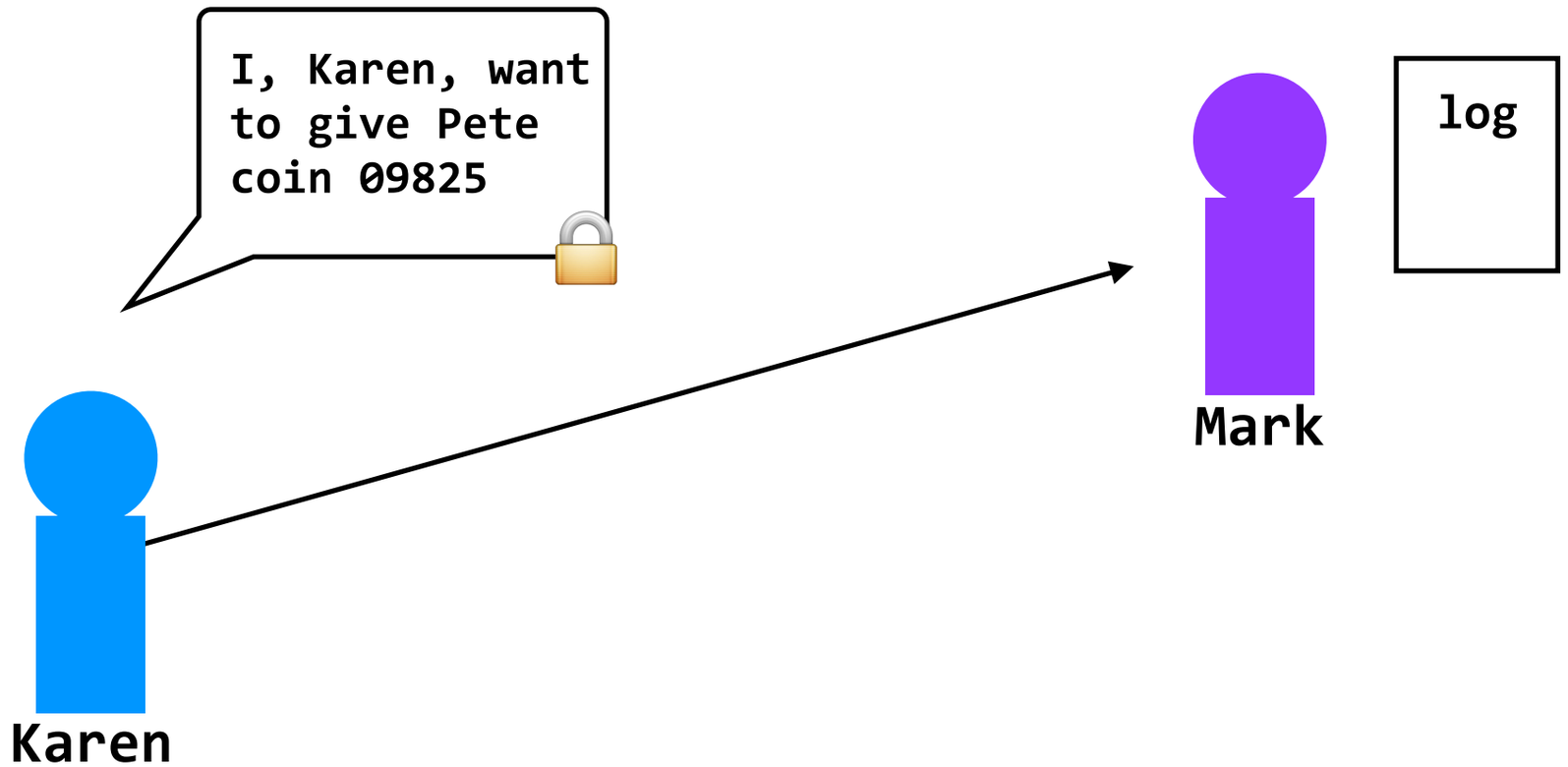
idea: get consensus from "enough" of the network — let's say 51% — before verifying the transaction



problem: Sybil Attacks

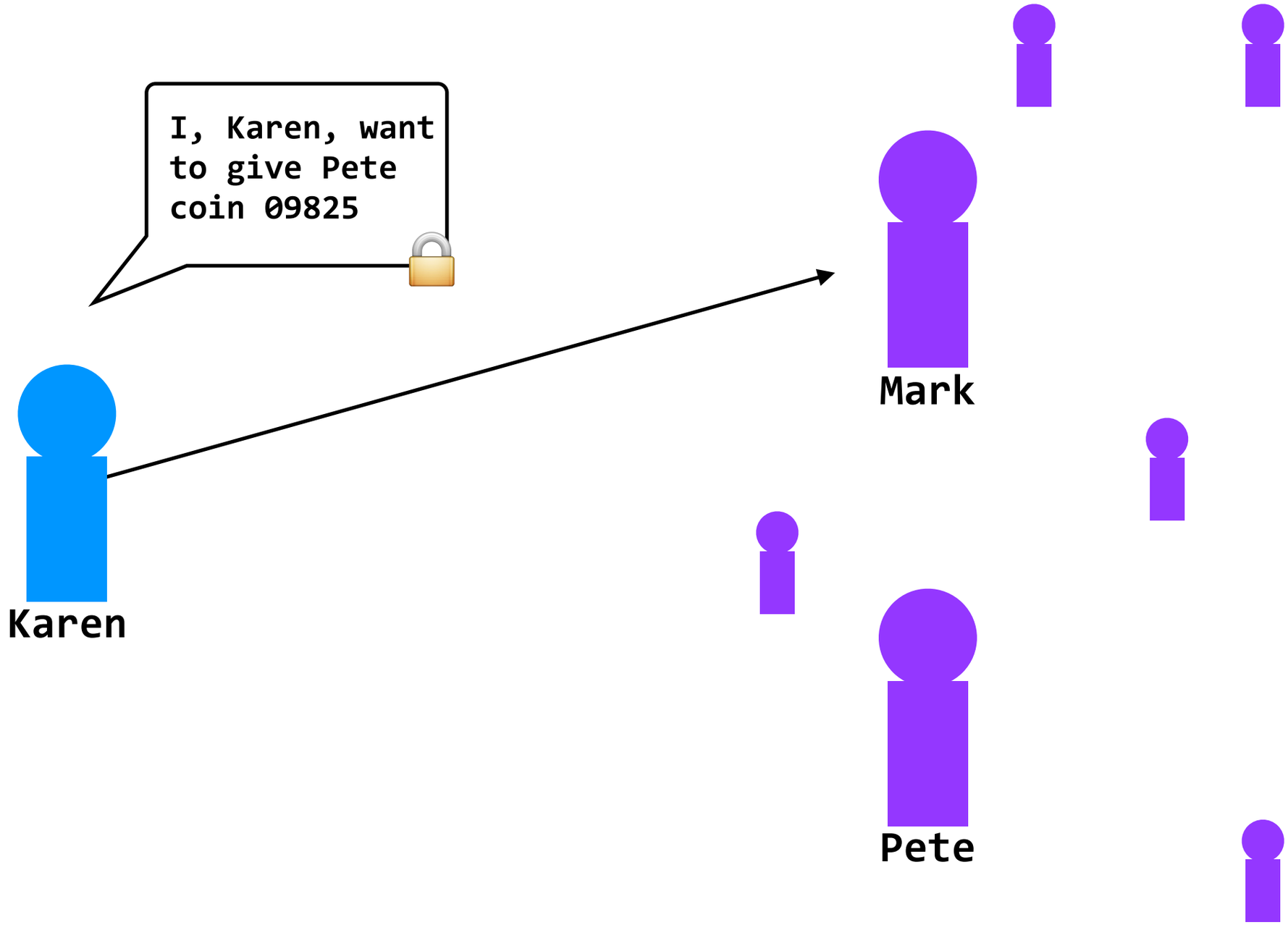
can be solved by using strong identities, but we want to be anonymous


I, Karen, want to give Pete coin 09825
Karen
Mark
Pete

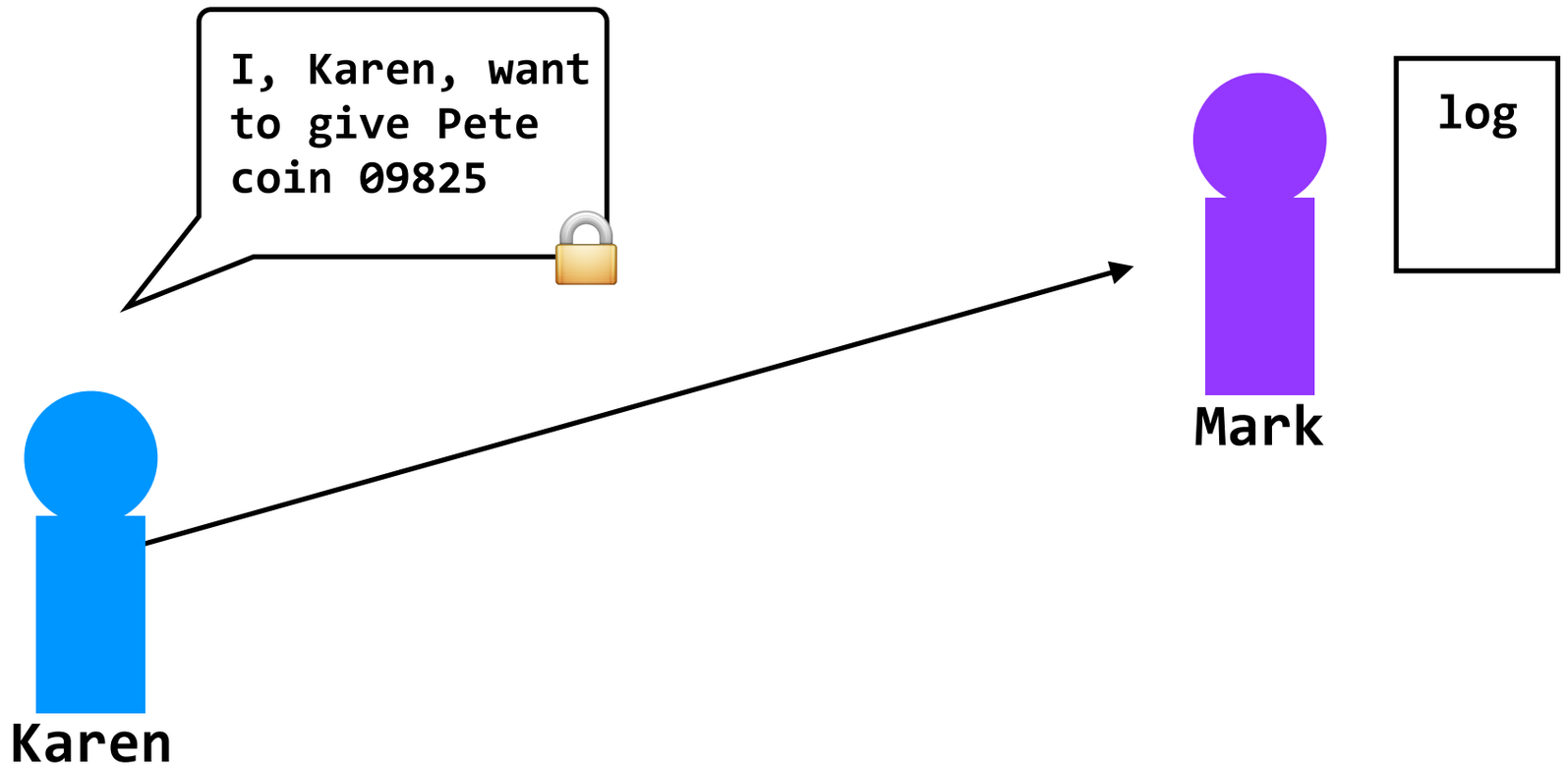


Mark uses his log to verify that Karen owns coin `09825`, and then sets about solving a **proof-of-work** to validate this transaction

Once he solves it, Mark broadcasts the transaction along with the solution to the rest of the network

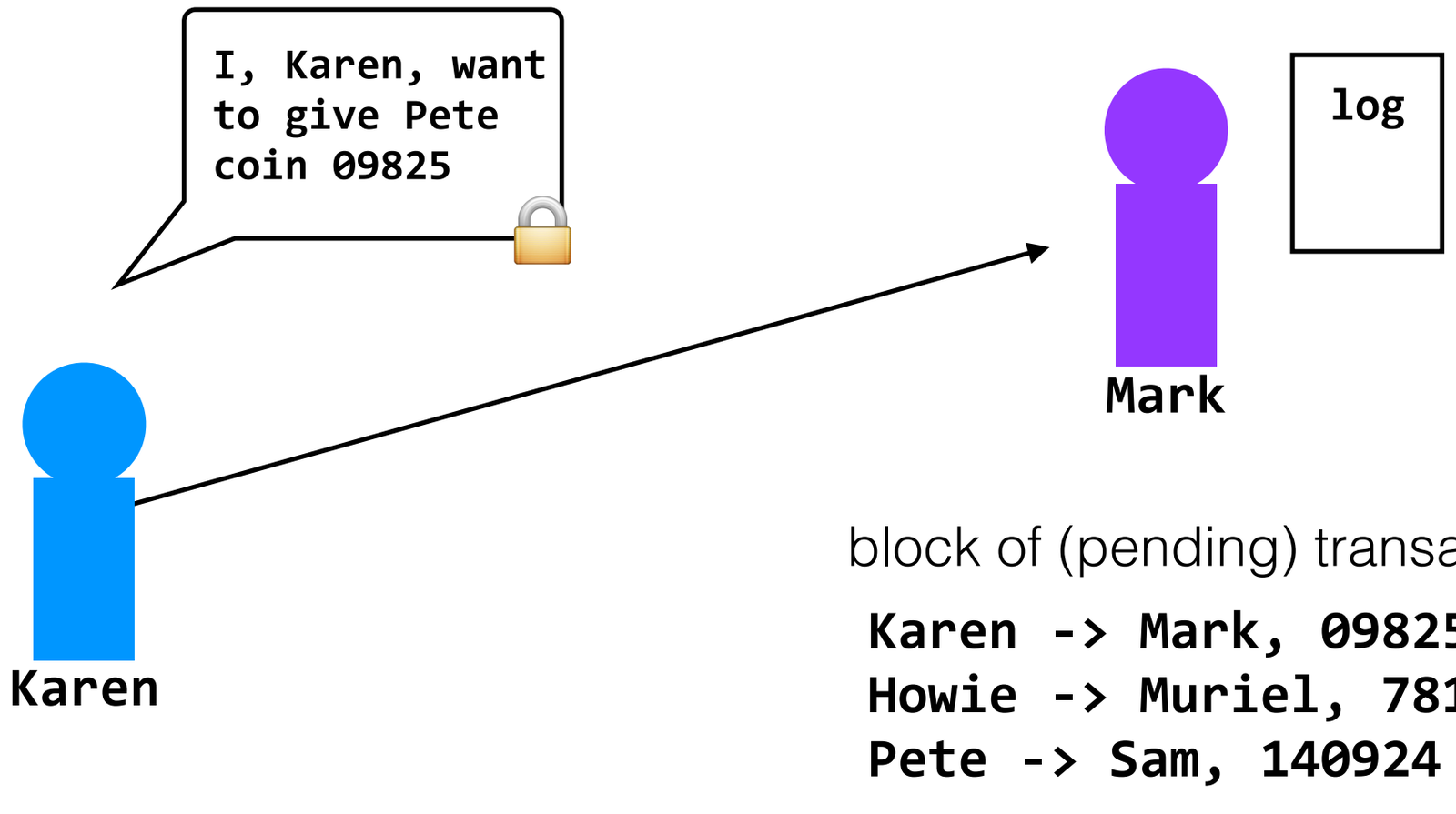


in reality, everyone in the network is competing to validate the transaction first



Mark uses his log to verify that Karen owns coin `09825`, and then sets about solving a **proof-of-work** to validate this transaction

Once he solves it, Mark broadcasts the transaction along with the solution to the rest of the network

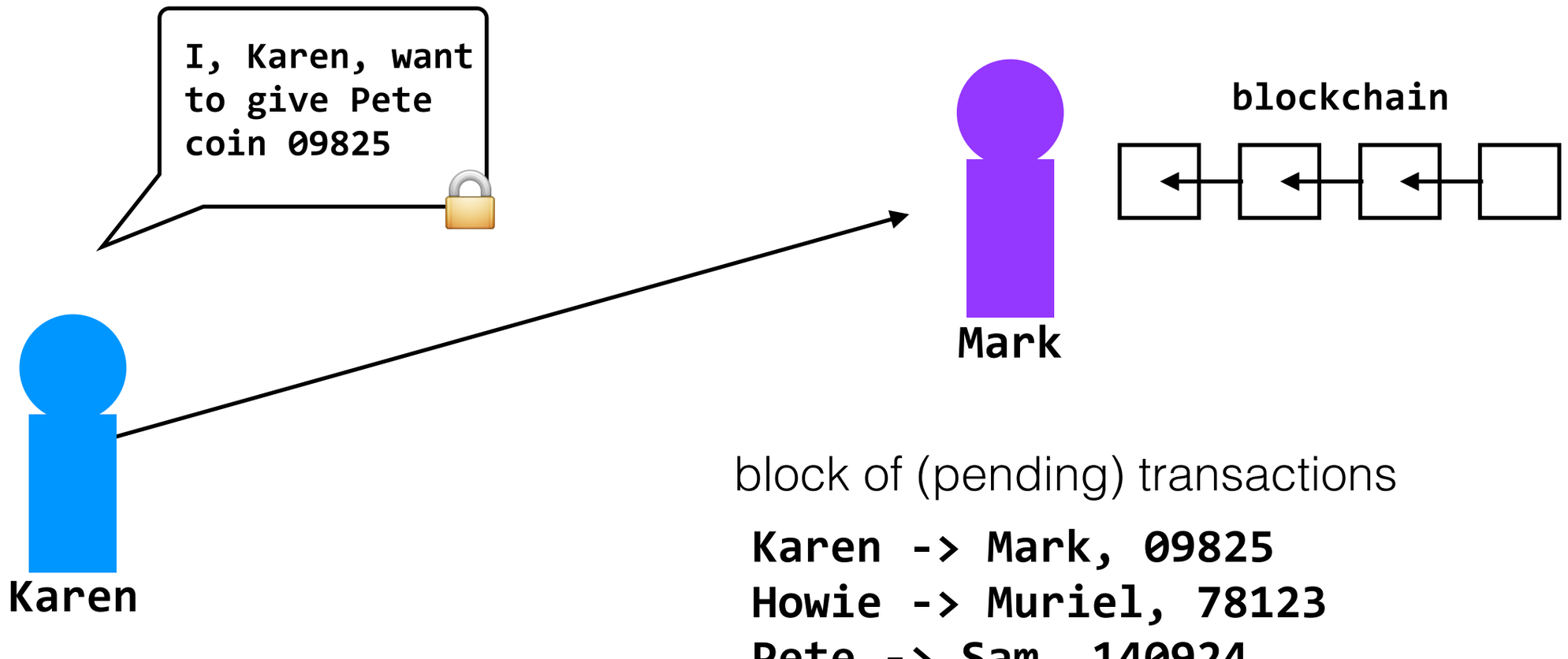


block of (pending) transactions

```
Karen -> Mark, 09825
Howie -> Muriel, 78123
Pete -> Sam, 140924
```

typically, users will verify a **block** of transactions at once, rather than one transaction at a time

since each block of transactions includes a hash that points to the previous block, we refer to the log as the **blockchain**

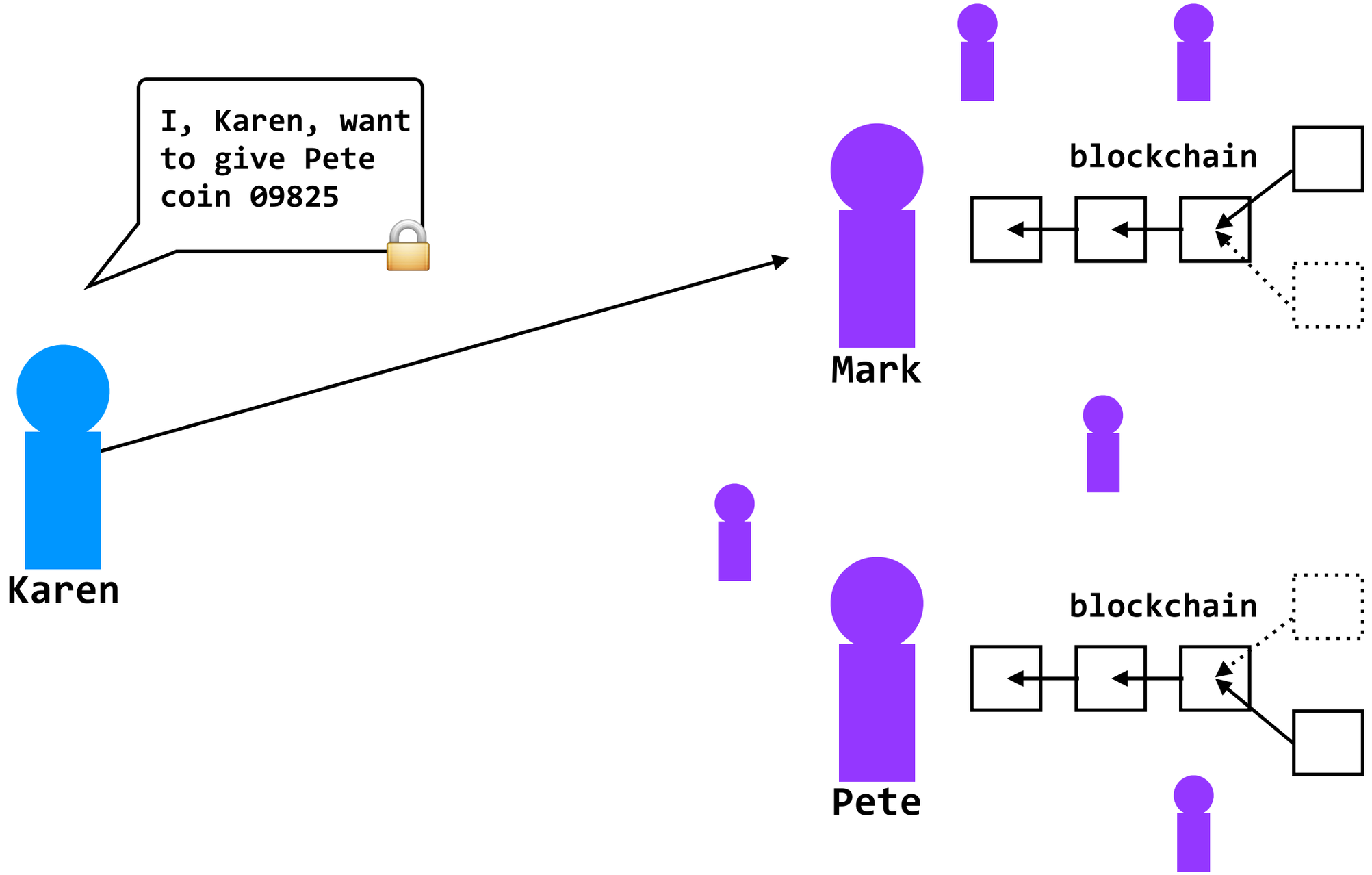


block of (pending) transactions

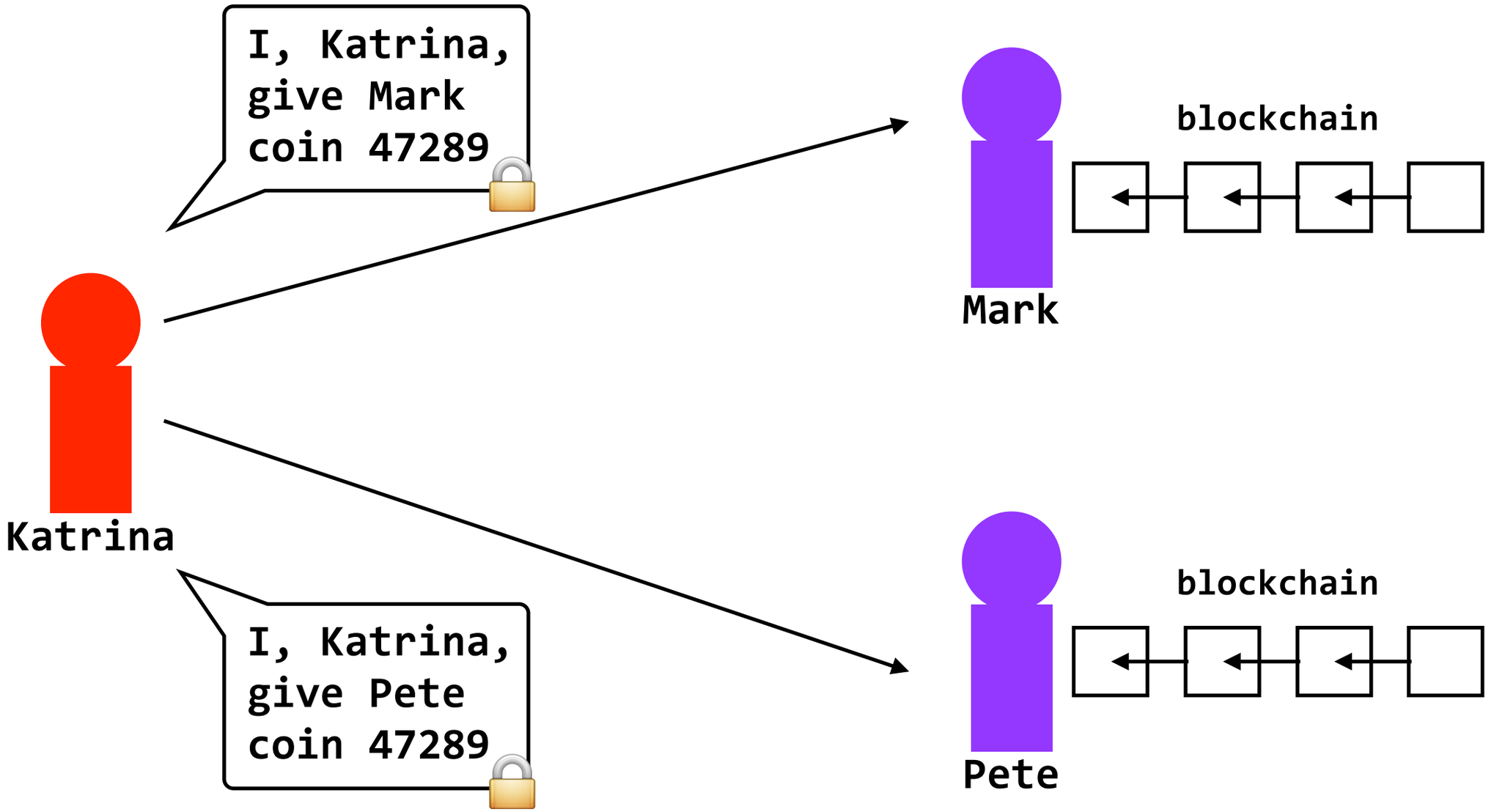
```
Karen -> Mark, 09825
Howie -> Muriel, 78123
Pete -> Sam, 140924
```

typically, users will verify a **block** of transactions at once, rather than one transaction at a time

since each block of transactions includes a hash that points to the previous block, we refer to the log as the **blockchain**

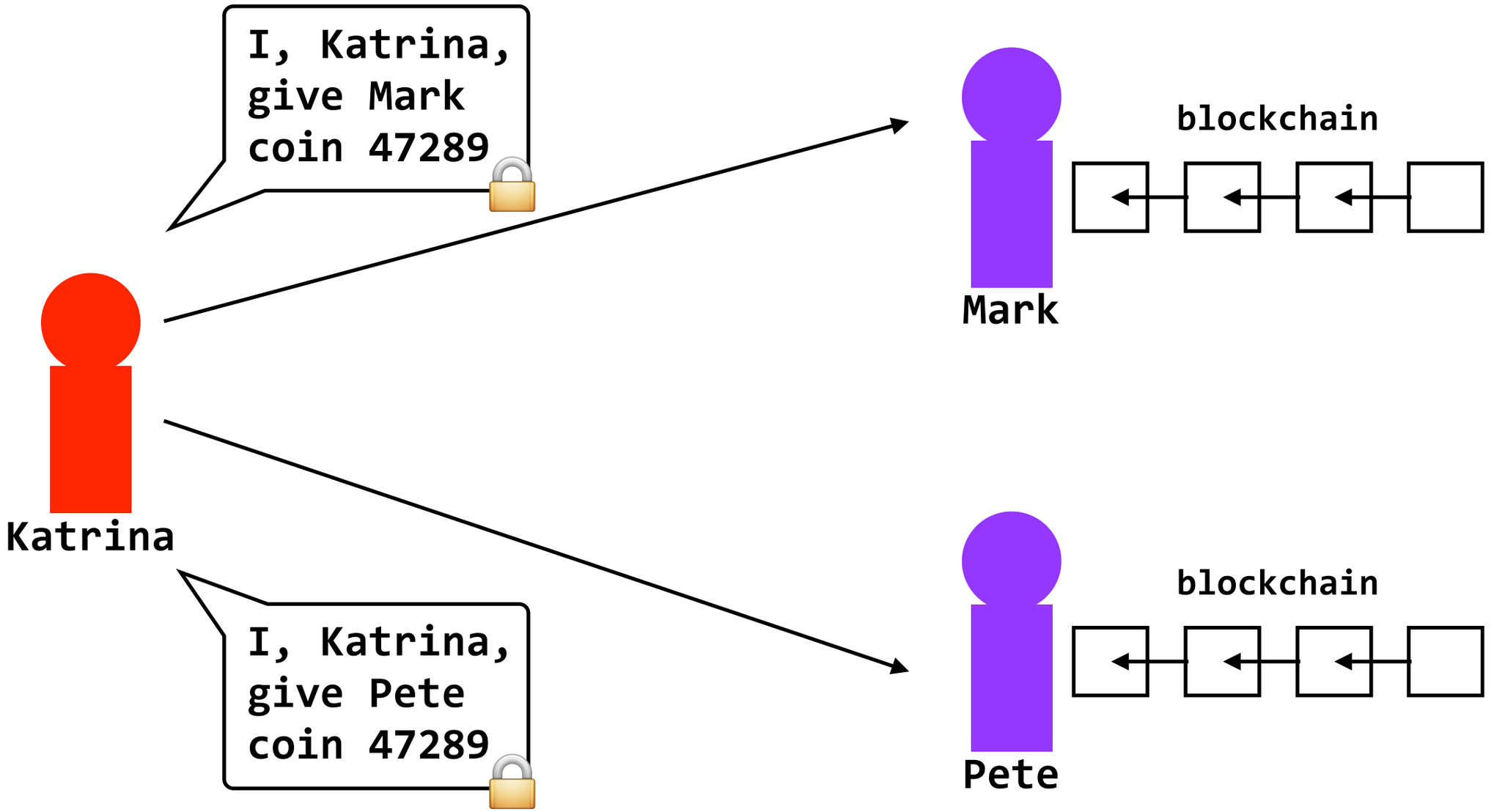


If multiple users “win” the competition at (roughly) the same time, the blockchain will **fork**. Bitcoin resolves this problem by having miners work only on the longest fork, quickly rendering the other branch obsolete



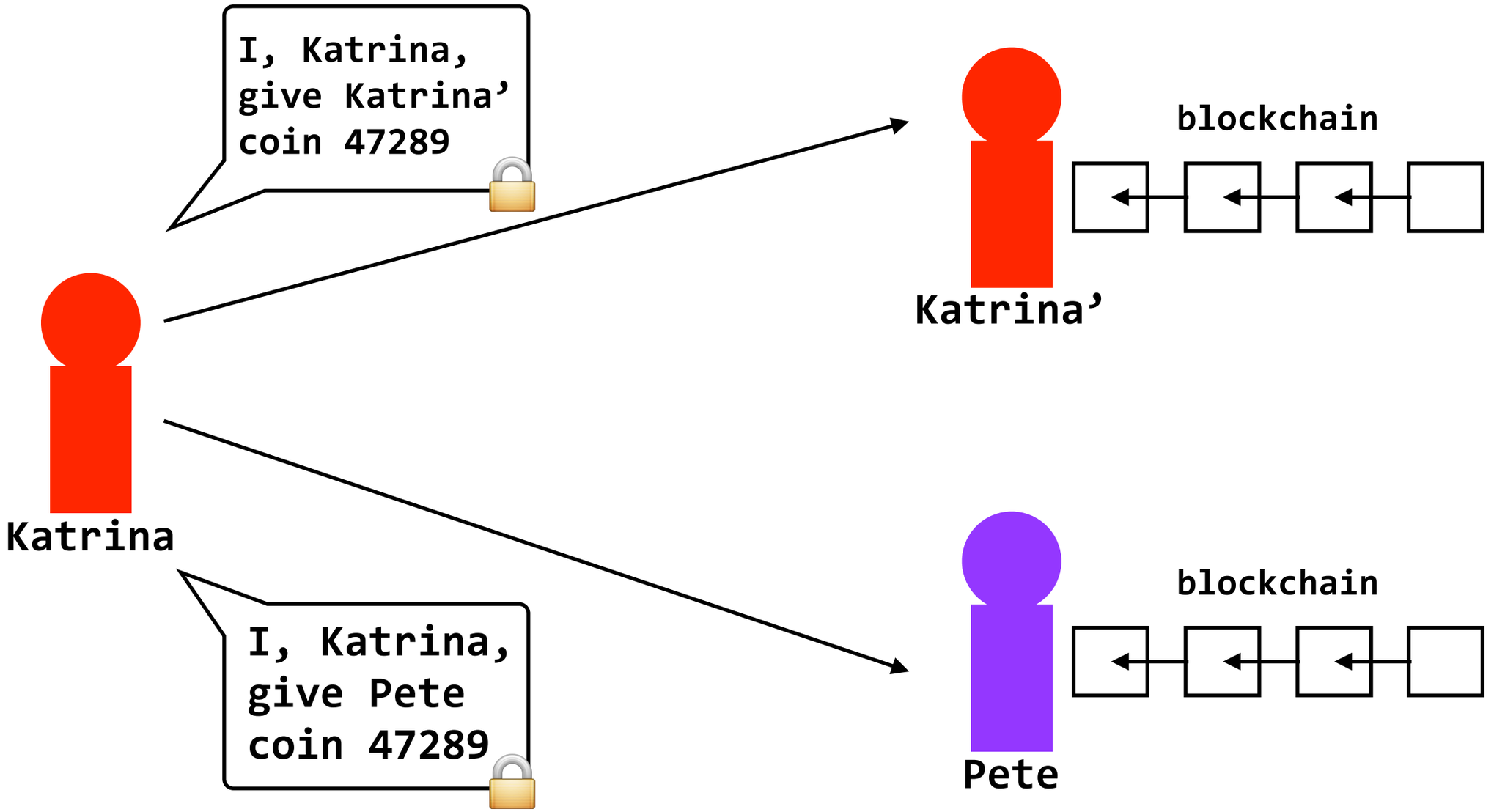
idea: Katrina tries to validate a block that includes both of those transactions

won't work — other users will examine the block and spot inconsistencies



idea: Katrina tries to get both transactions validated on the network

won't work — eventually network will confirm only one



idea: Katrina tries to spend a coin with Pete and herself (Katrina' is a Sybil of Katrina)

Katrina would need a lot of compute power to pull this off

- **Bitcoin** is a decentralized digital currency. Being decentralized means that there is no bank; in Bitcoin, everyone is the bank.

- Bitcoin provides a distributed public log called the **blockchain** that can be used for purposes other than digital currency. It uses **proofs-of-work** to prevent Sybil Attacks, since strong identities won't work.

- In theory, users of Bitcoin are **anonymous**; in practice, it's not clear how true that is.