6.033 Spring 2017

Lecture #24

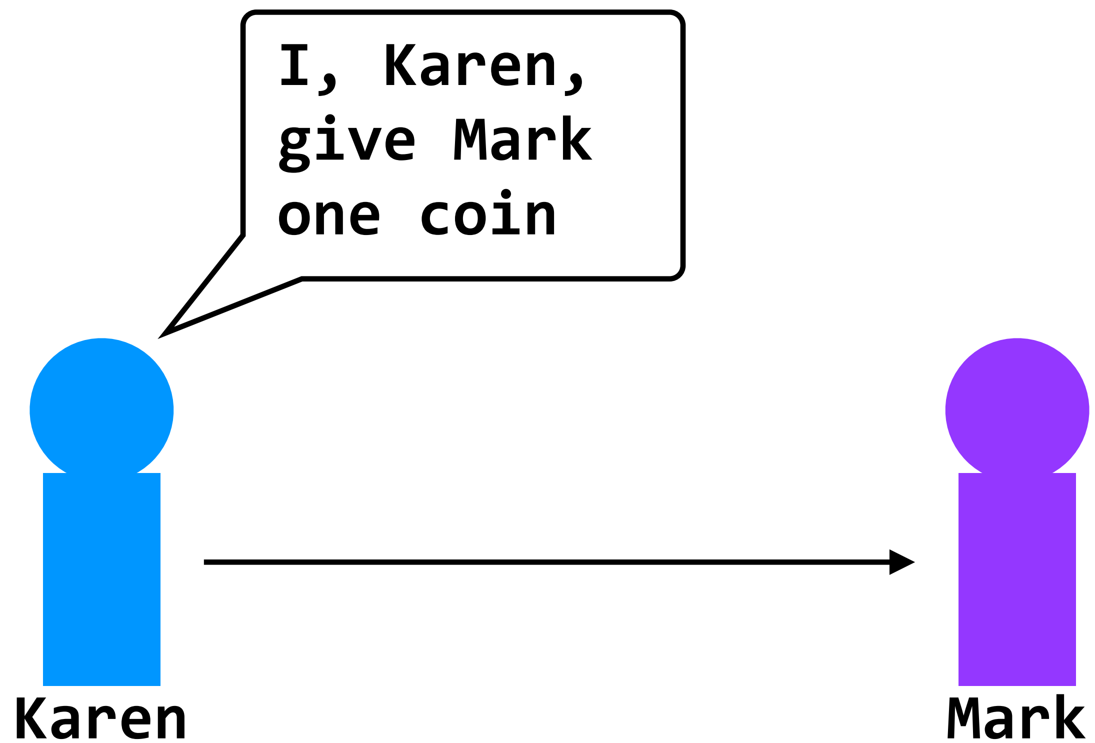
- **Anonymity and Digital Currency**

Bitcoin and Tor

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

decentralized currency

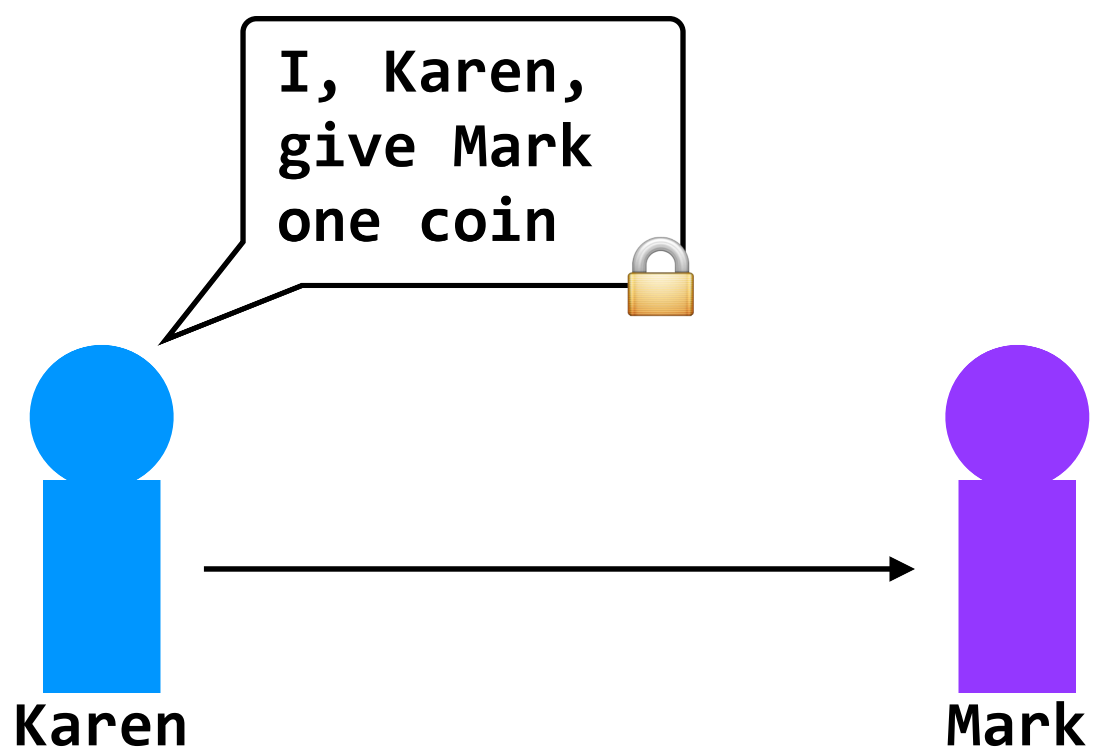
can we avoid having a centralized bank?



problem: easily forgeable

decentralized currency

can we avoid having a centralized bank?



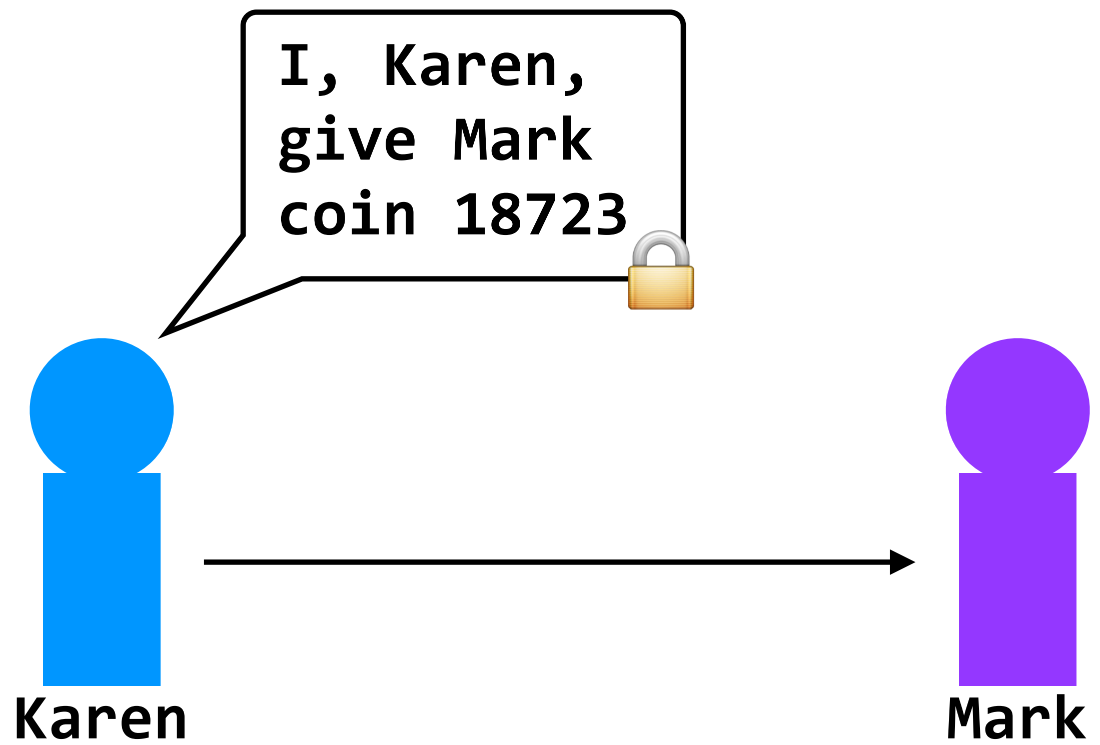
use digital signatures

Karen signs the message with her secret key

problem: replay attacks

decentralized currency

can we avoid having a centralized bank?

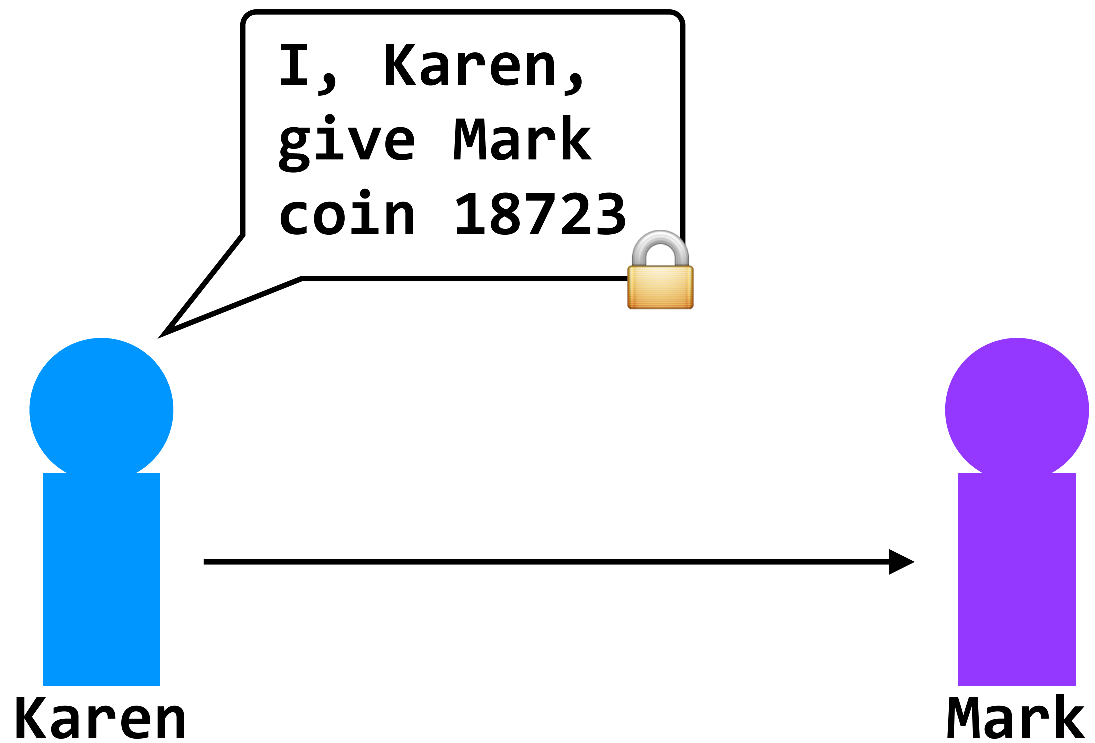


use sequence numbers

(serial numbers)

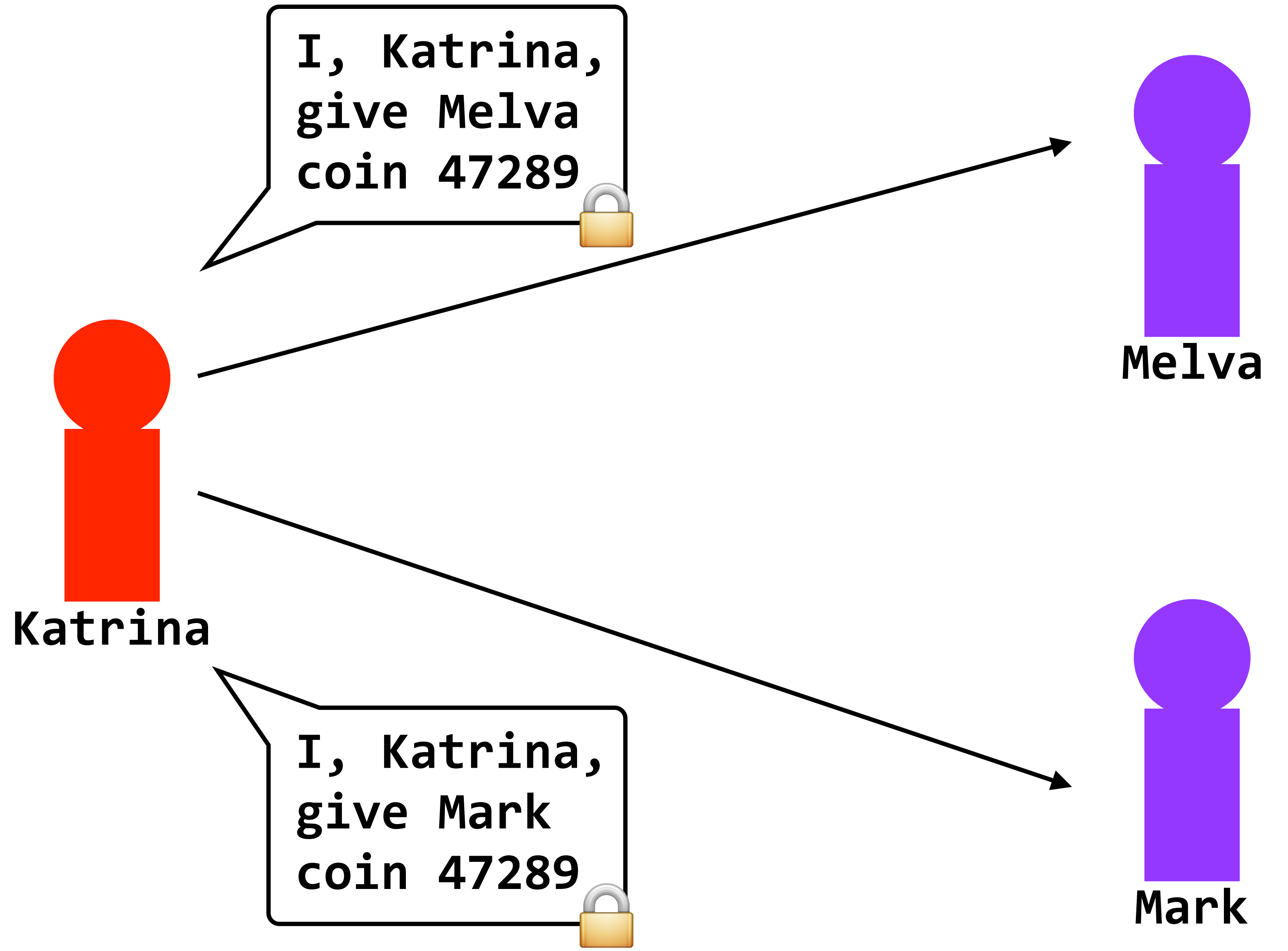
decentralized currency

can we avoid having a centralized bank?



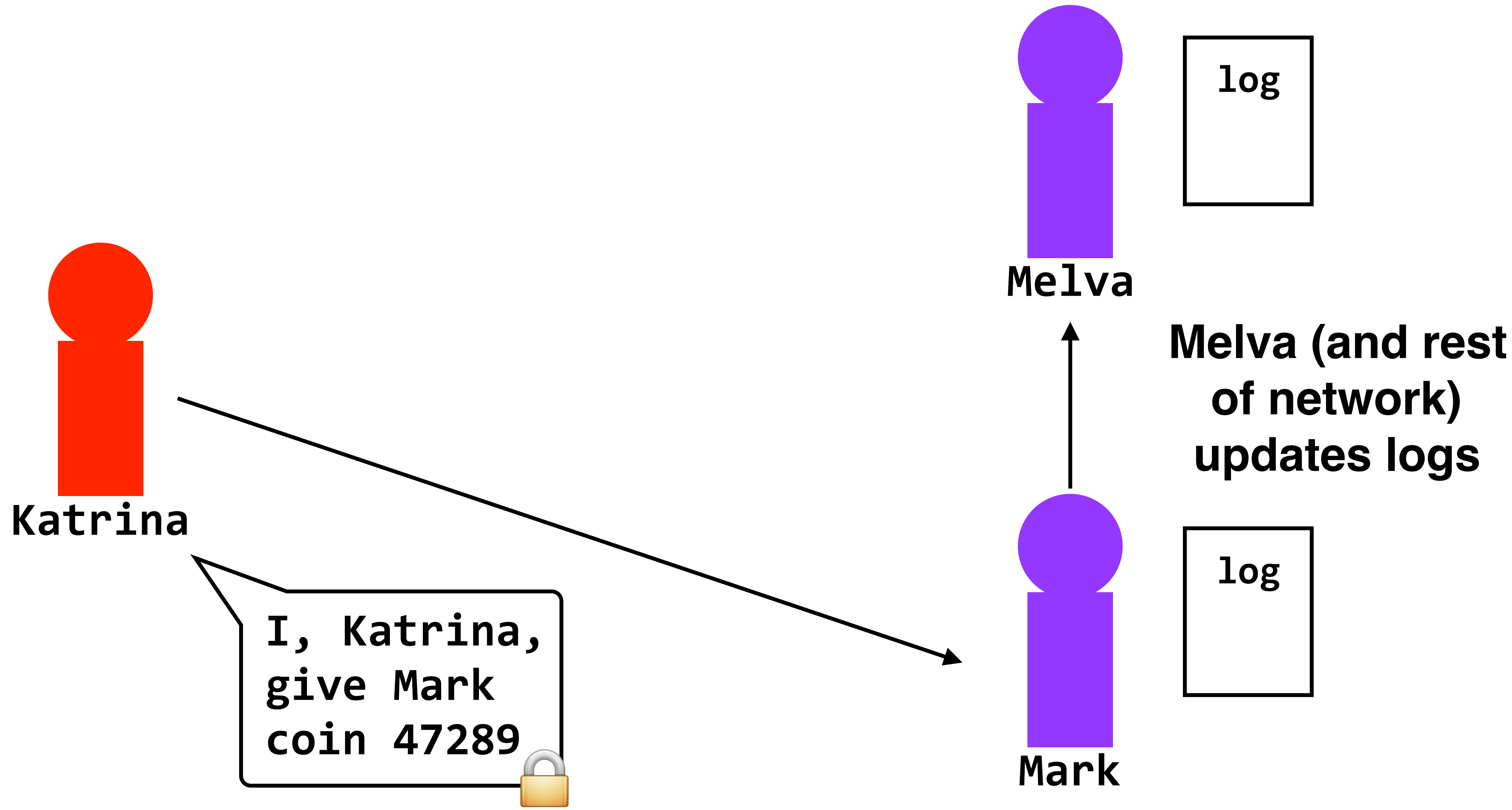
main technical challenge: how do we prevent double-spending?

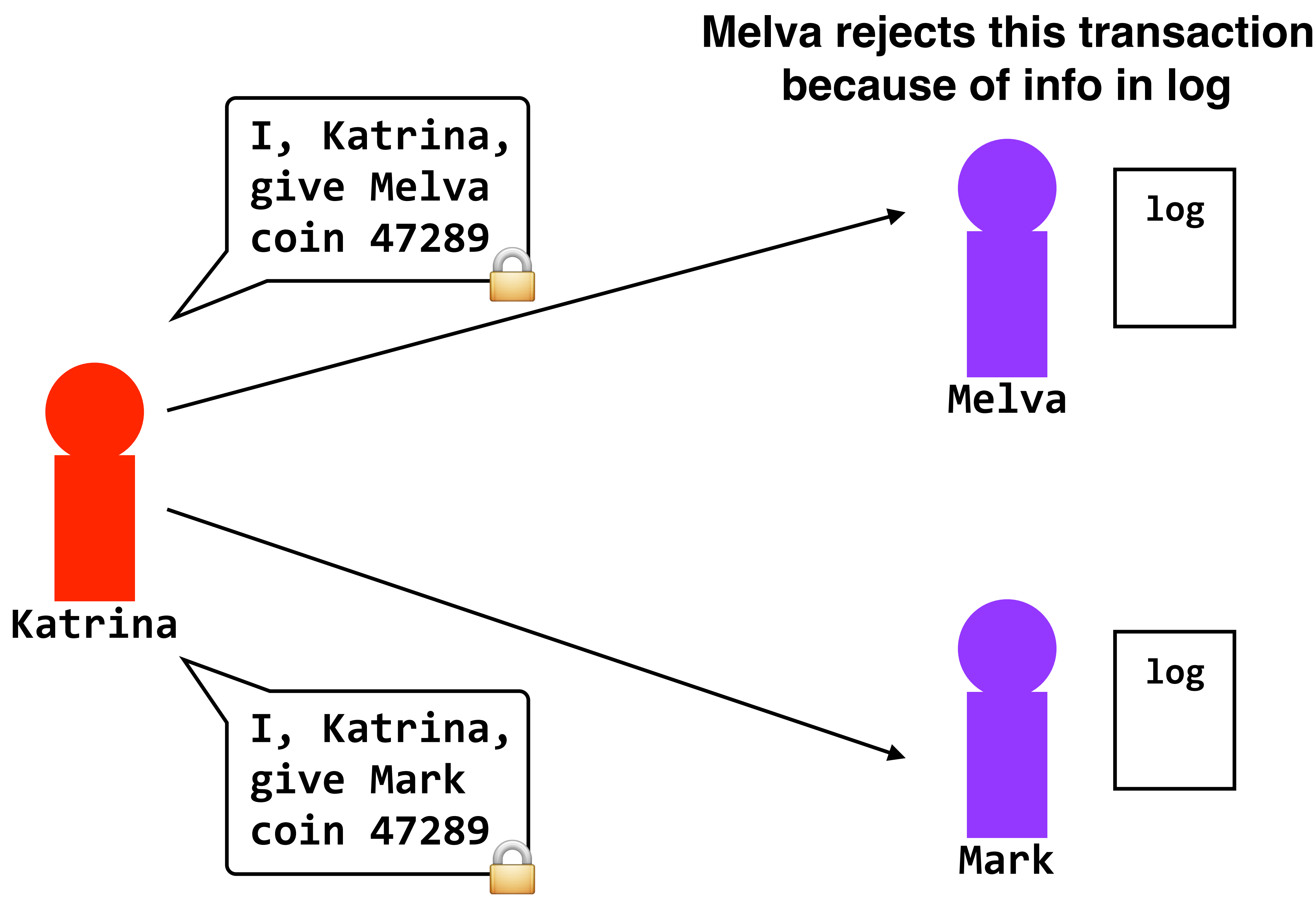
(other technical challenges will fall into place as we solve this)



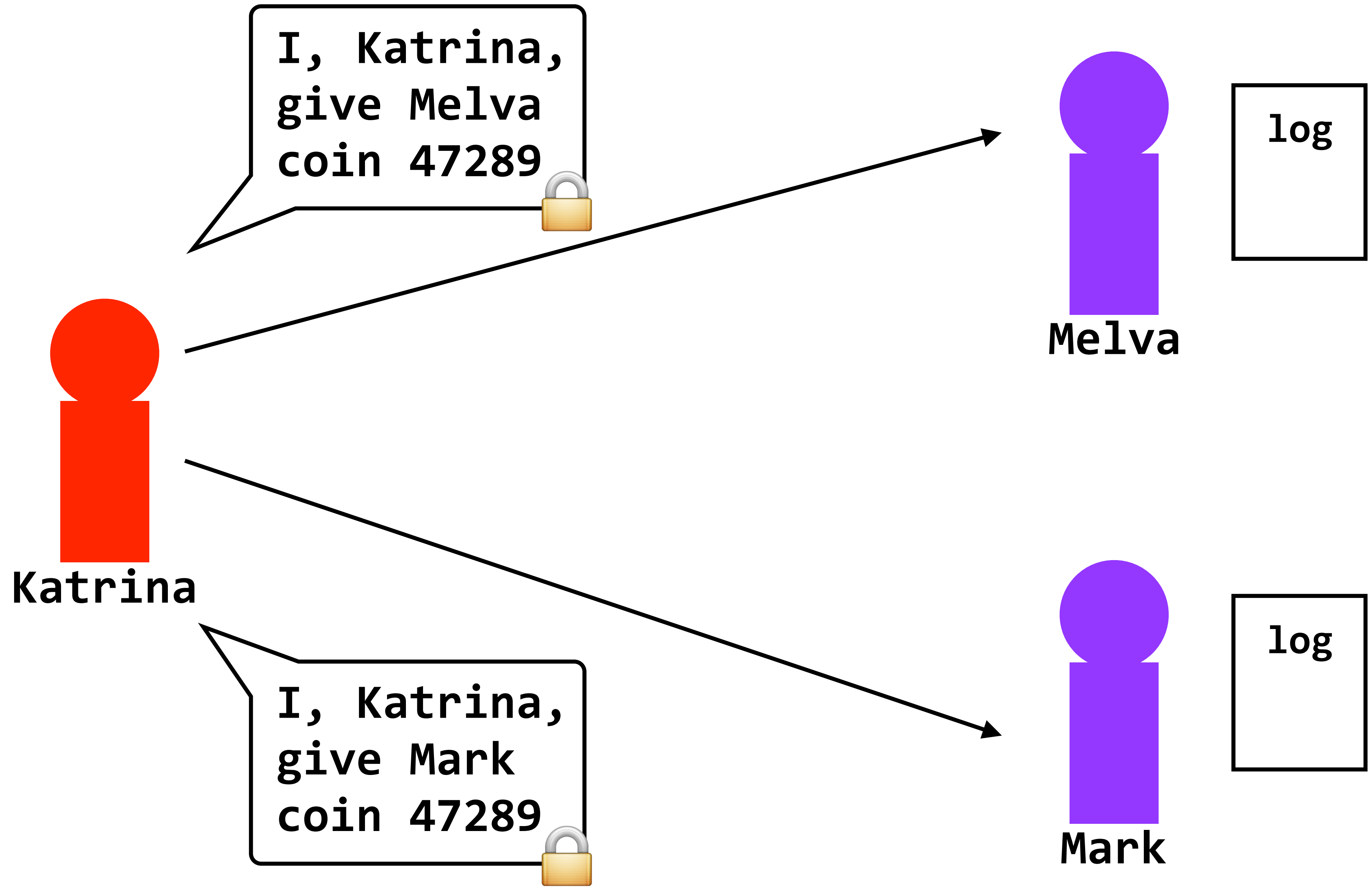
main technical challenge: how do we prevent double-spending?

(other technical challenges will fall into place as we solve this)

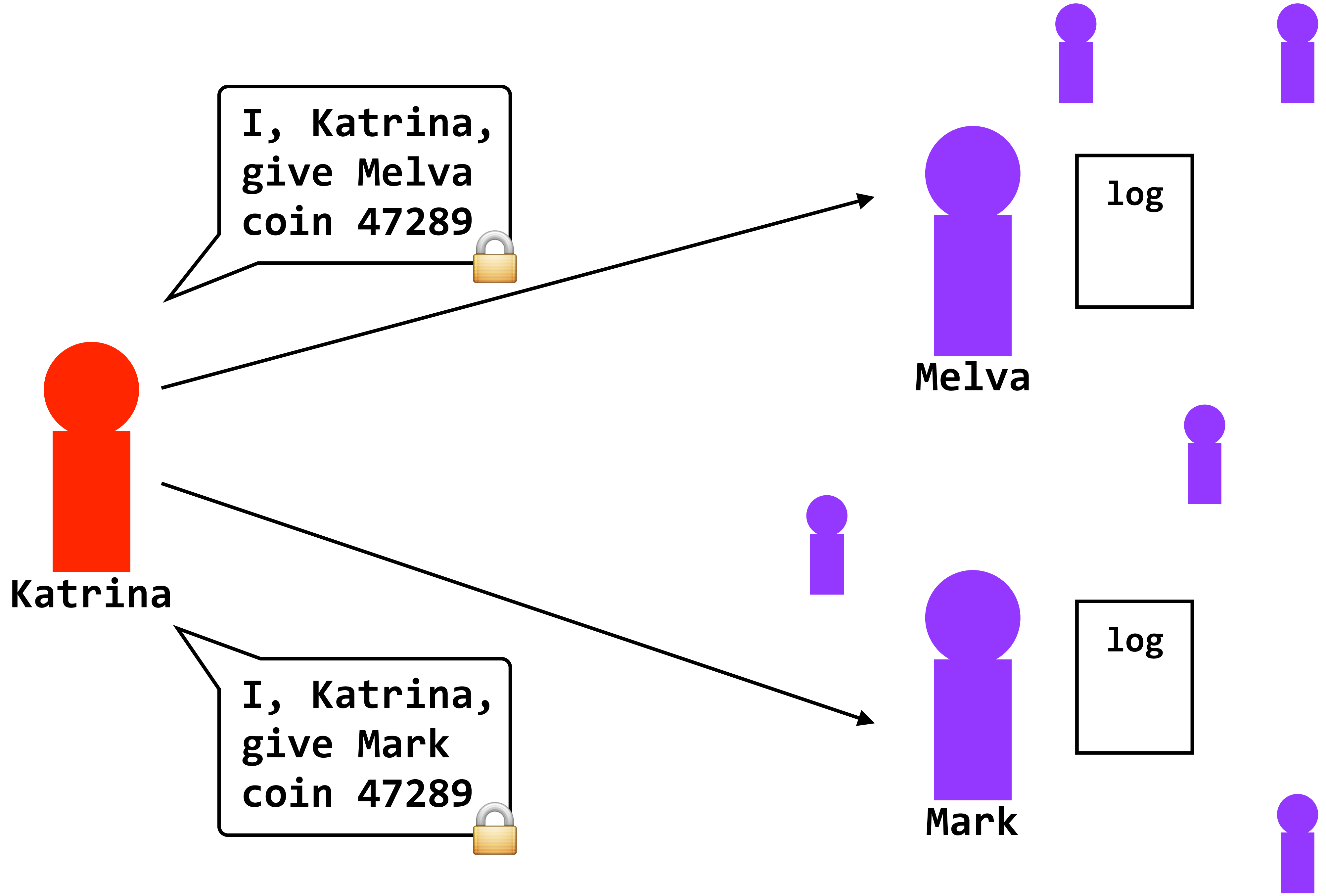

log
Melva
Melva (and rest of network) updates logs
Katrina
I, Katrina, give Mark coin 47289
log
Mark



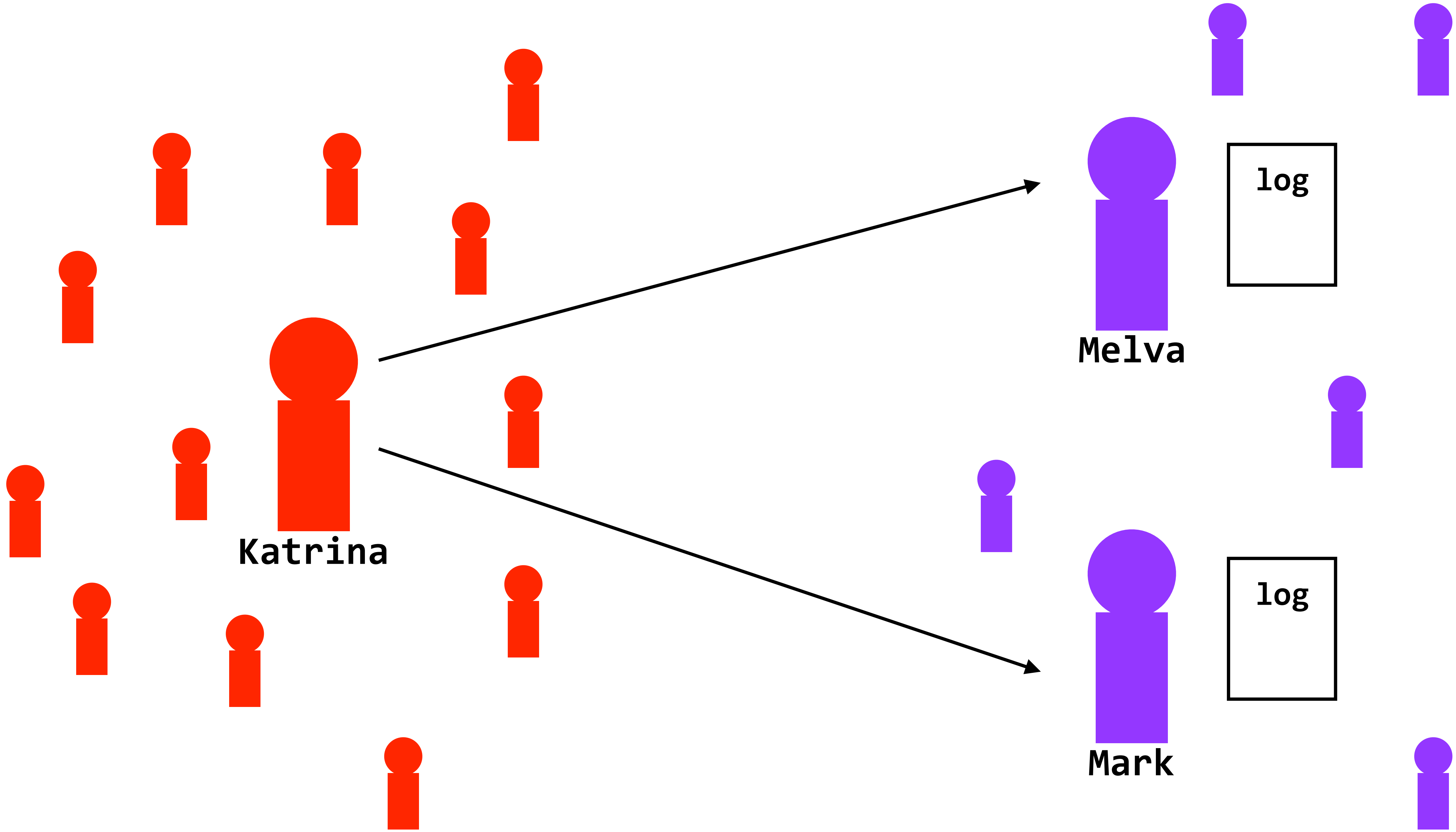
problem: what if Katrina tries to spend with Mark and Melva at the same time? (before either party has a chance to publish the transaction)



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

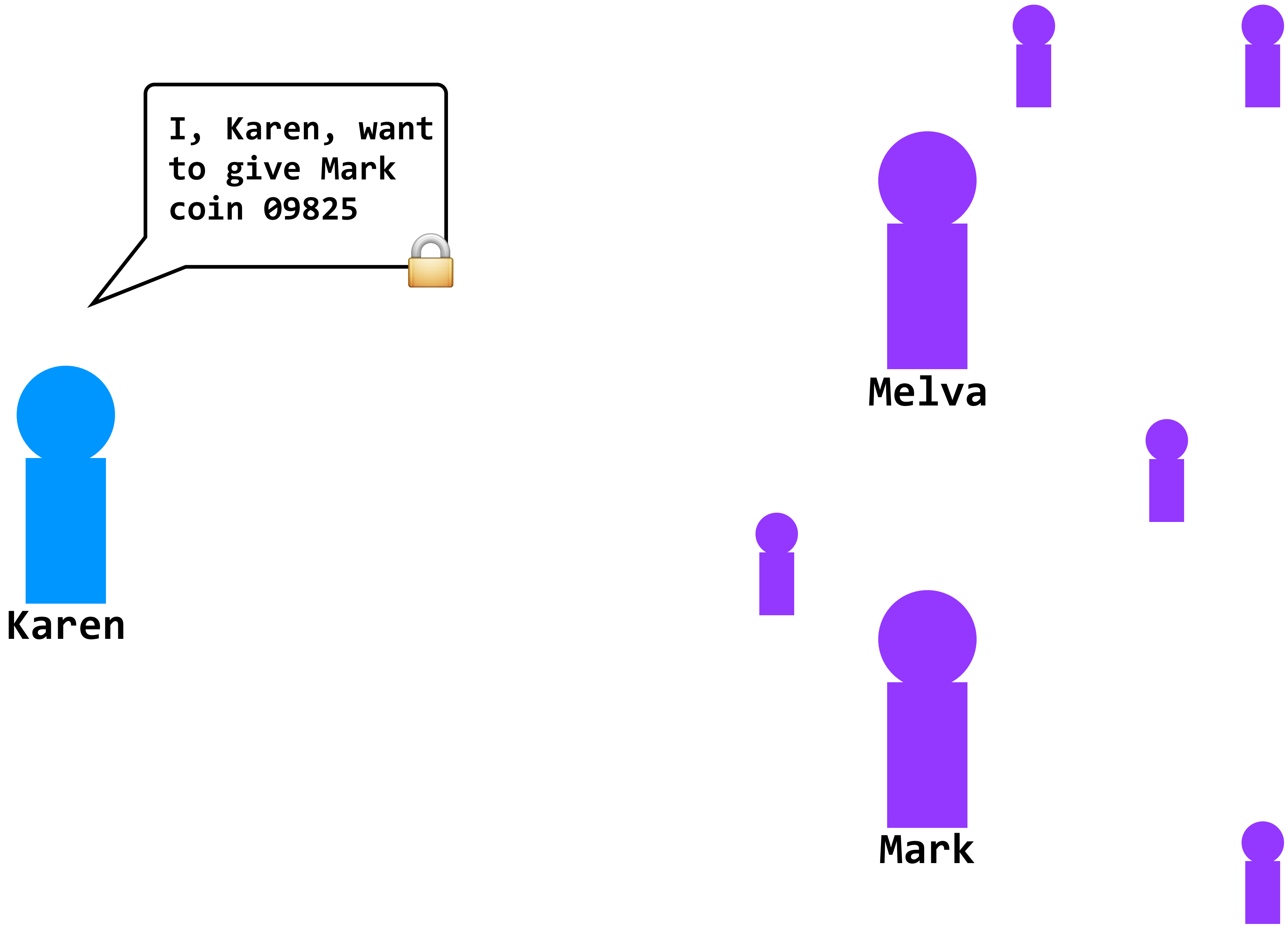


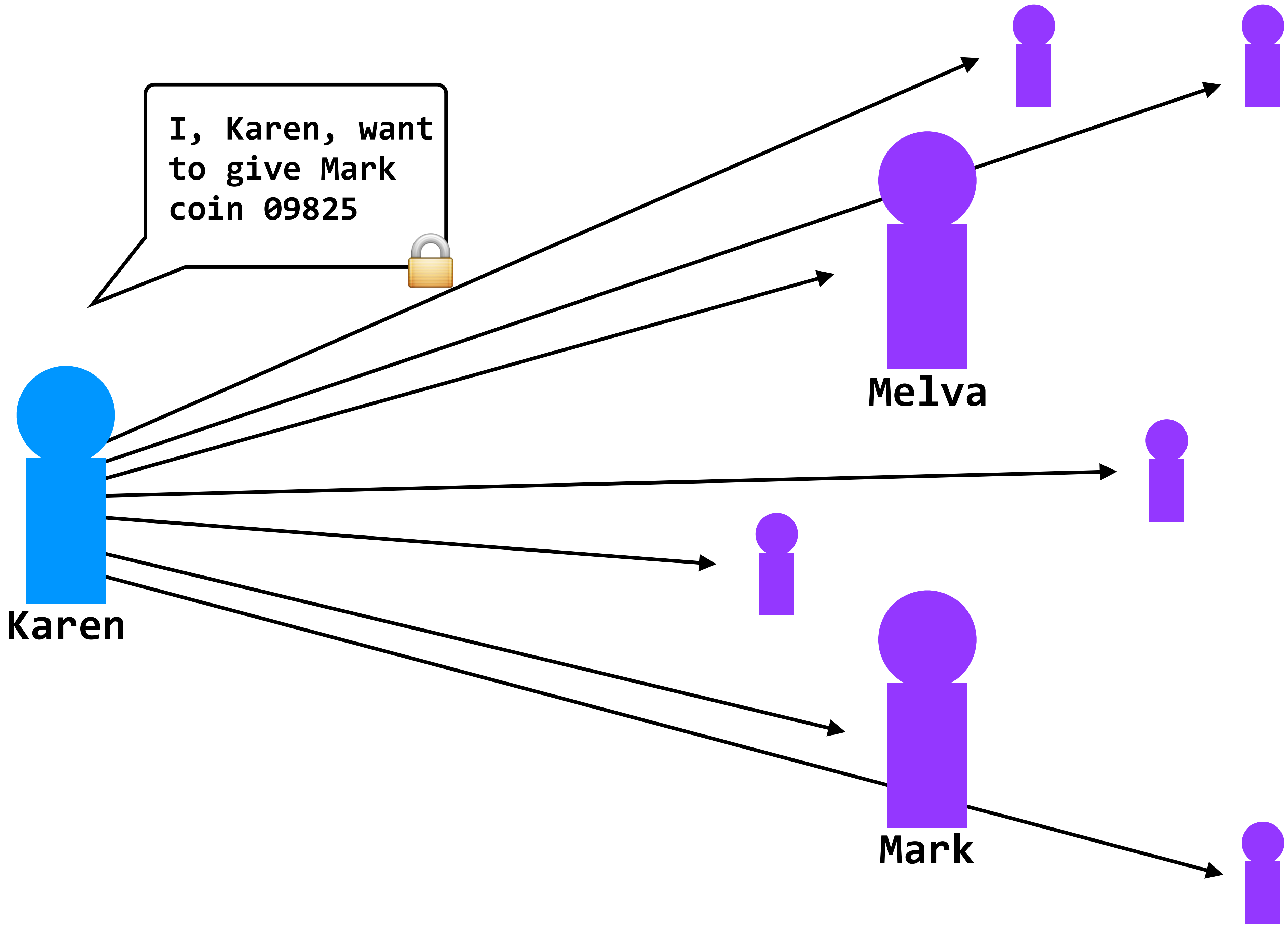
idea: get consensus from "enough" of the network — let's say 51% — before verifying the transaction

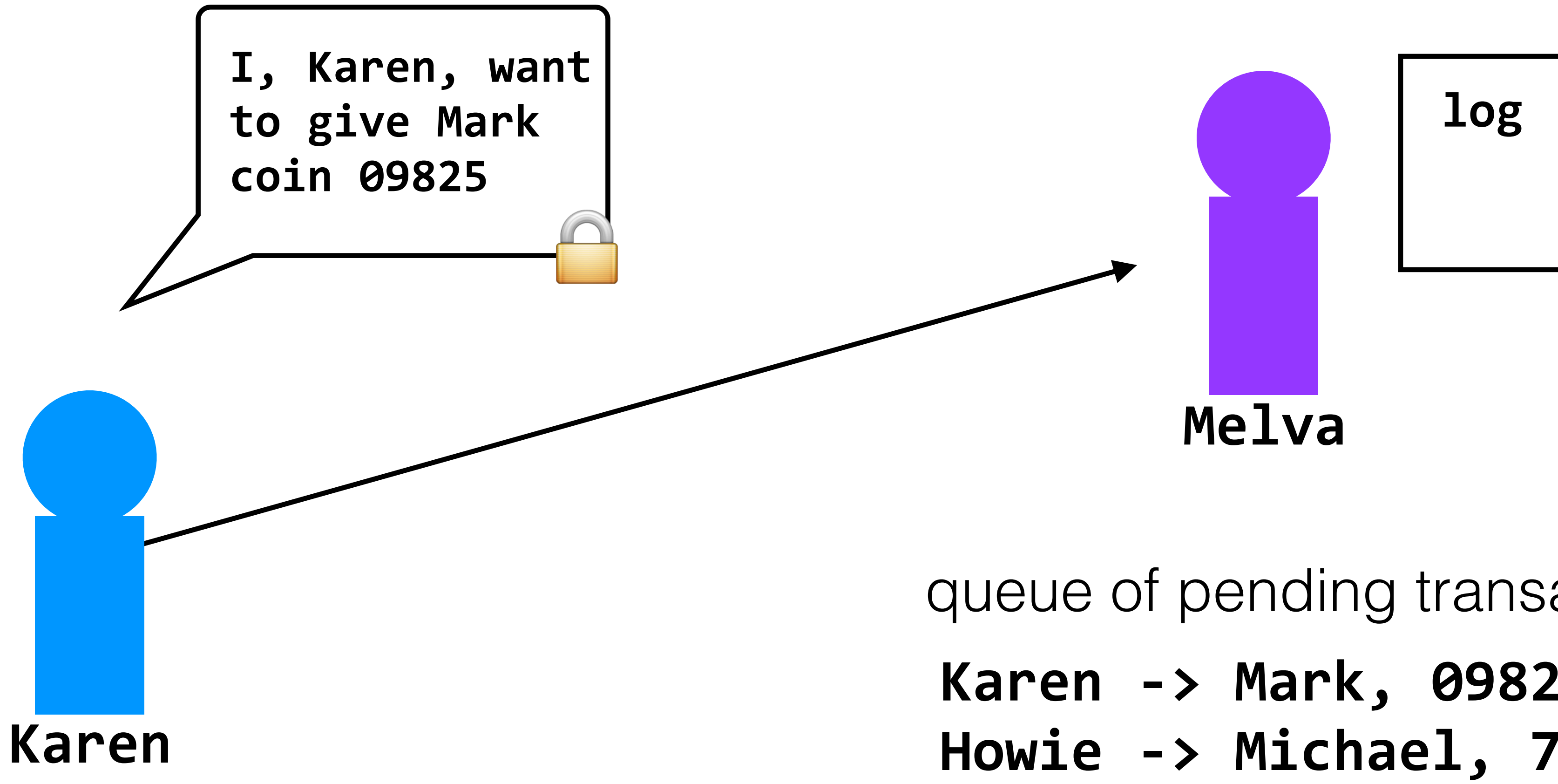


problem: Sybil Attacks

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

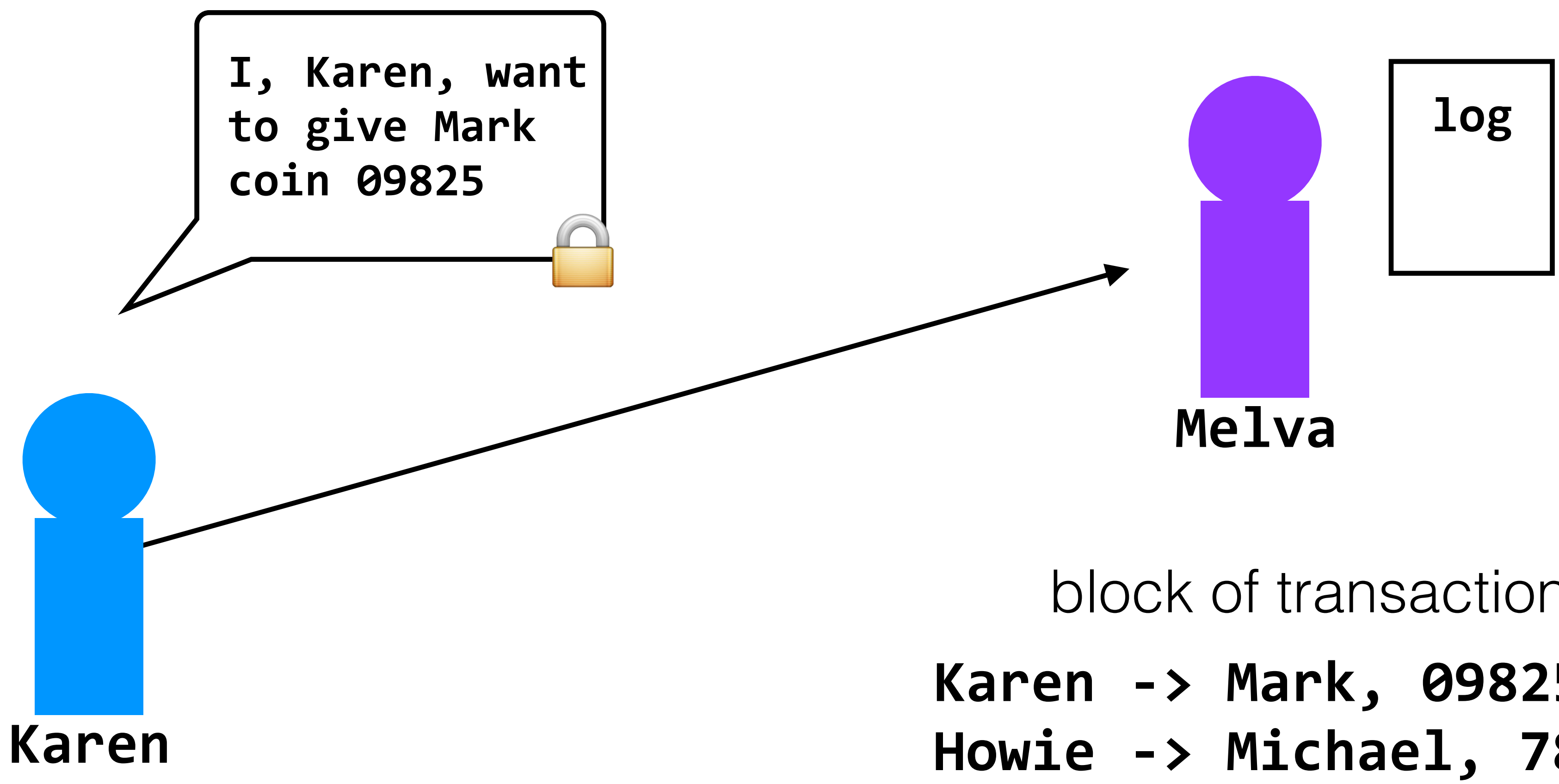

I, Karen, want to give Mark coin 09825
Karen
Melva
Mark


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



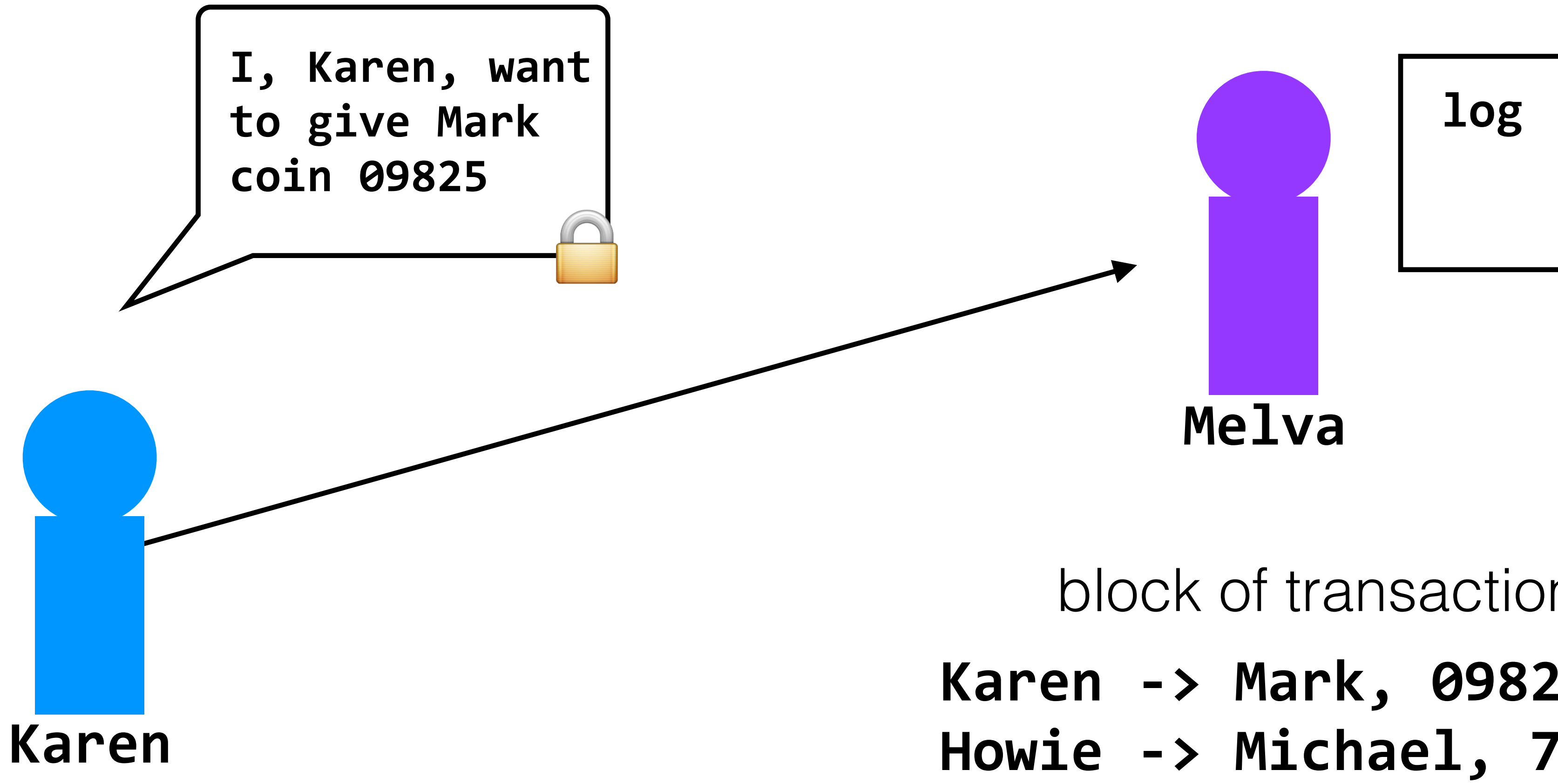
queue of pending transactions

```
Karen -> Mark, 09825
Howie -> Michael, 78123
Pete -> Jing, 140924
```



block of transactions

Karen -> Mark, 09825
Howie -> Michael, 78123
Pete -> Jing, 140924

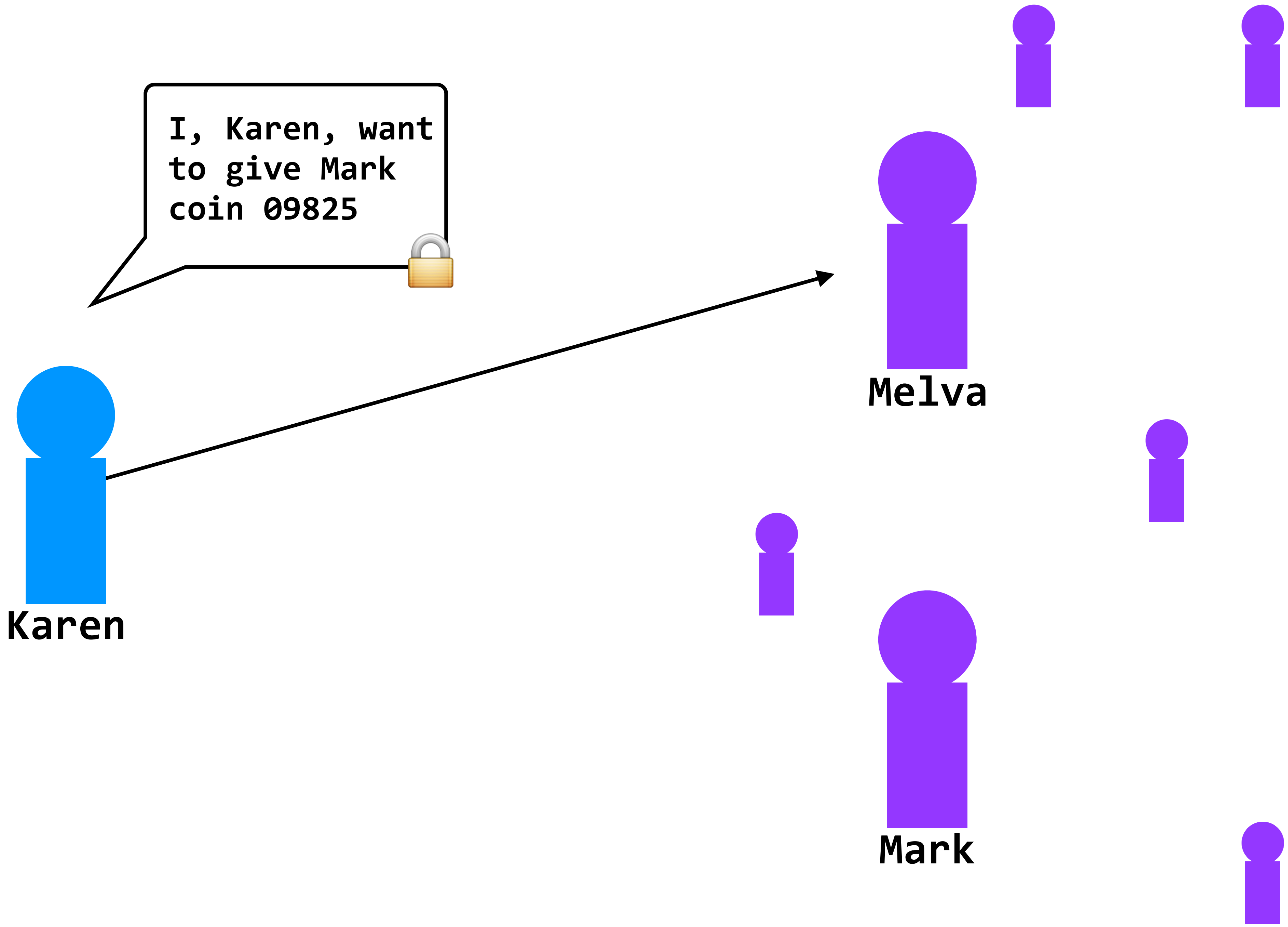


block of transactions

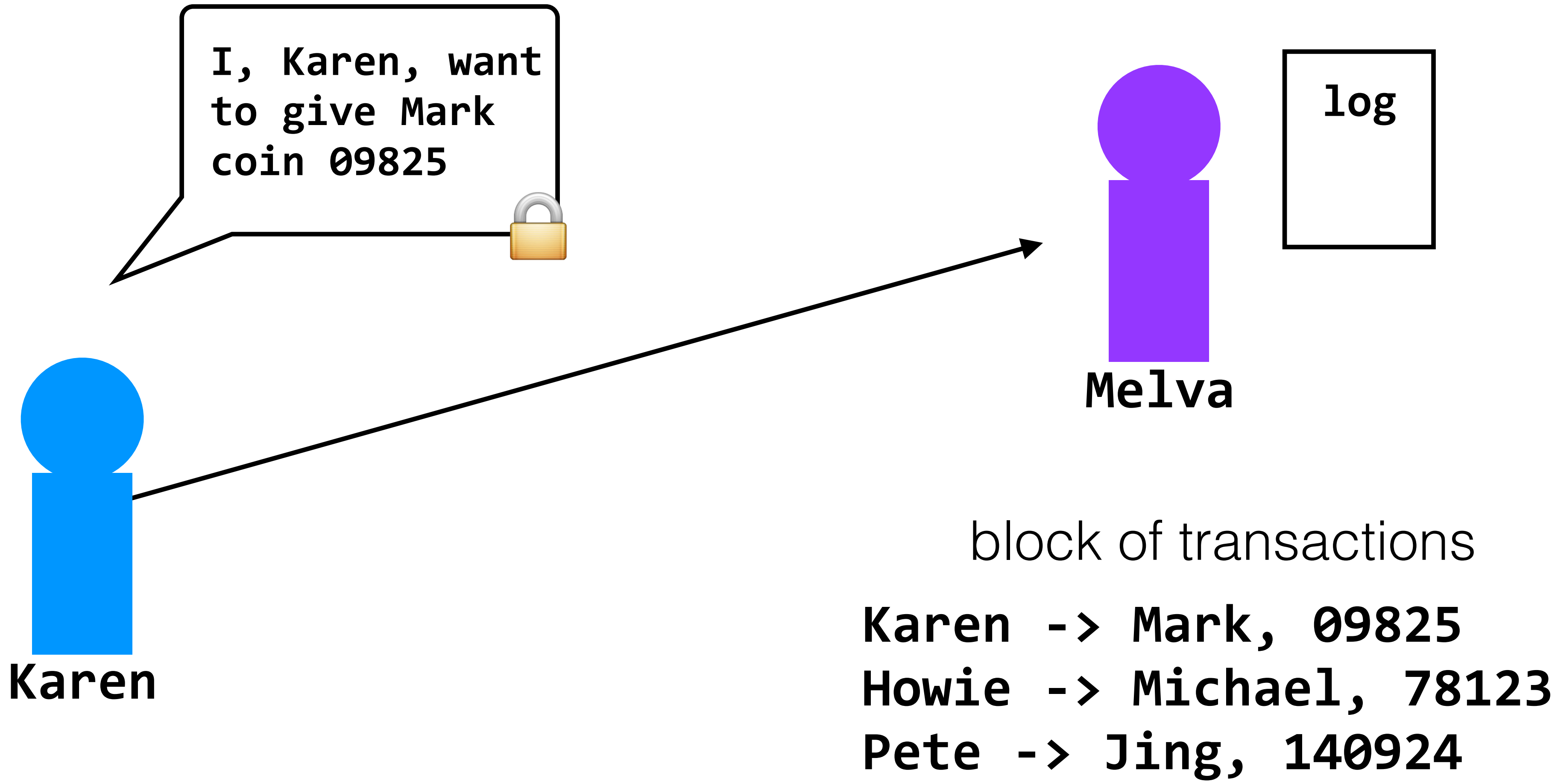
```
Karen -> Mark, 09825
Howie -> Michael, 78123
Pete -> Jing, 140924
```

Melva uses her log to verify that Karen owns coin `09825`, and then sets about solving a **proof-of-work** to validate this block of transactions

Once she solves it, Melva broadcasts the block along with the solution to the rest of the network, and gets a monetary reward

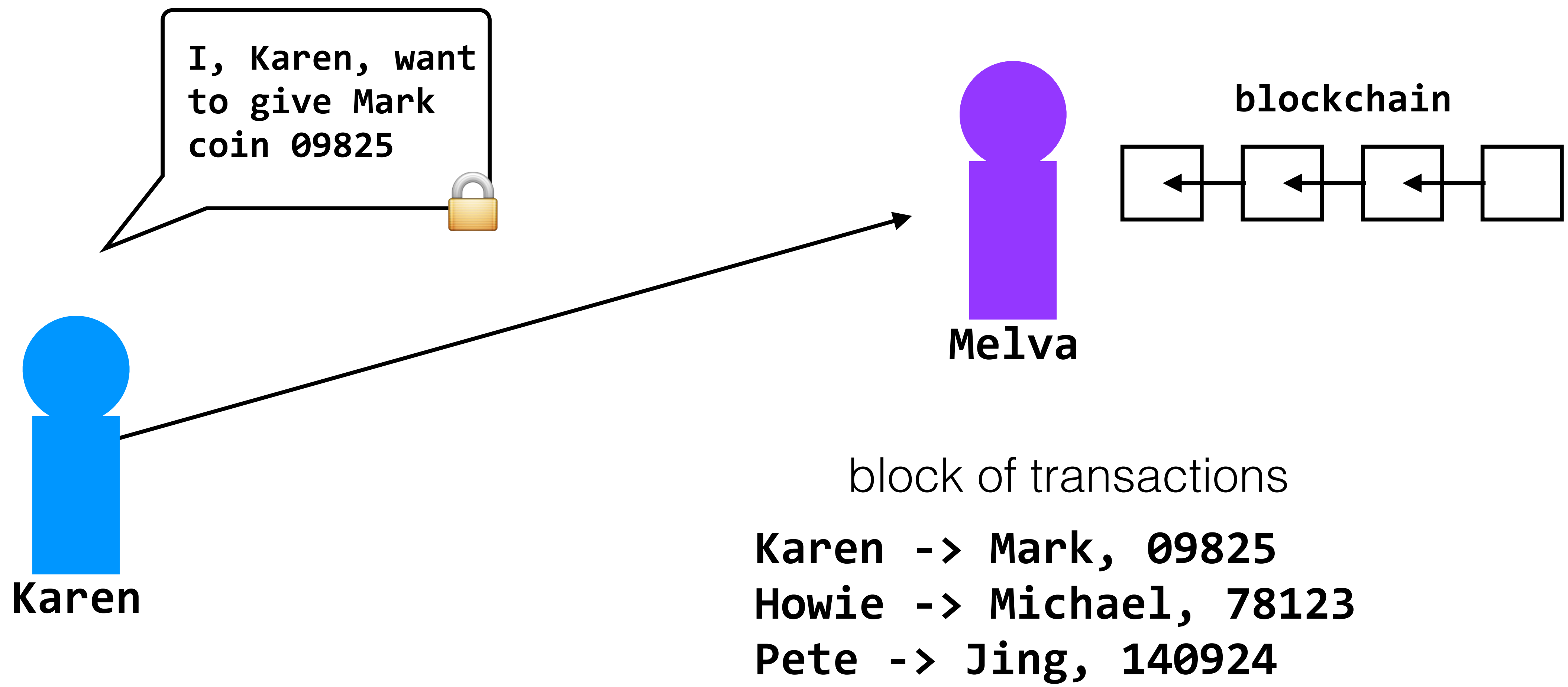


In reality, everyone in the network is competing to validate the transaction first (and receive the reward)



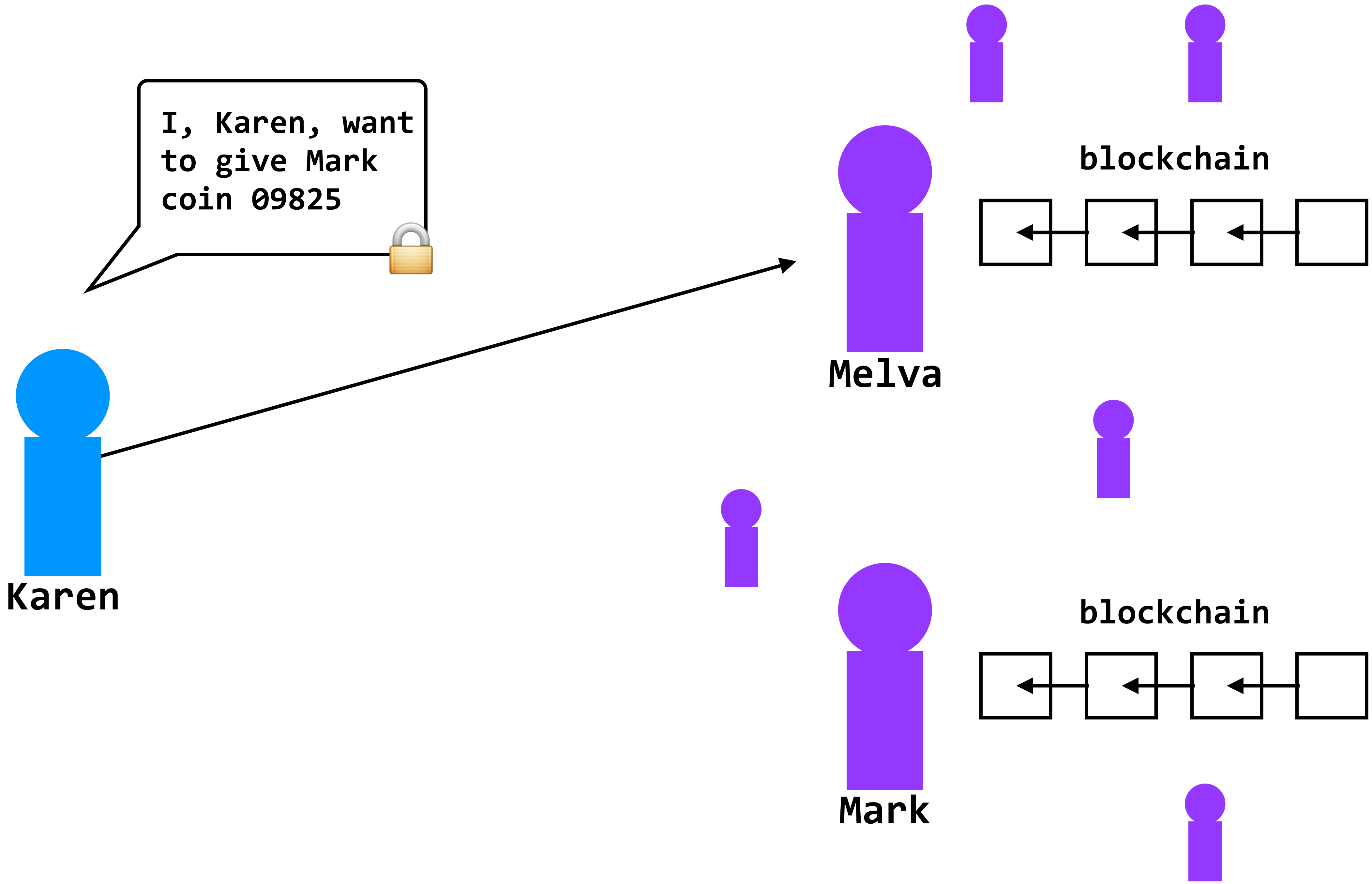
Melva uses her log to verify that Karen owns coin 09825, and then sets about solving a **proof-of-work** to validate this block of transactions

Once she solves it, Melva broadcasts the block along with the solution to the rest of the network, and gets a monetary reward

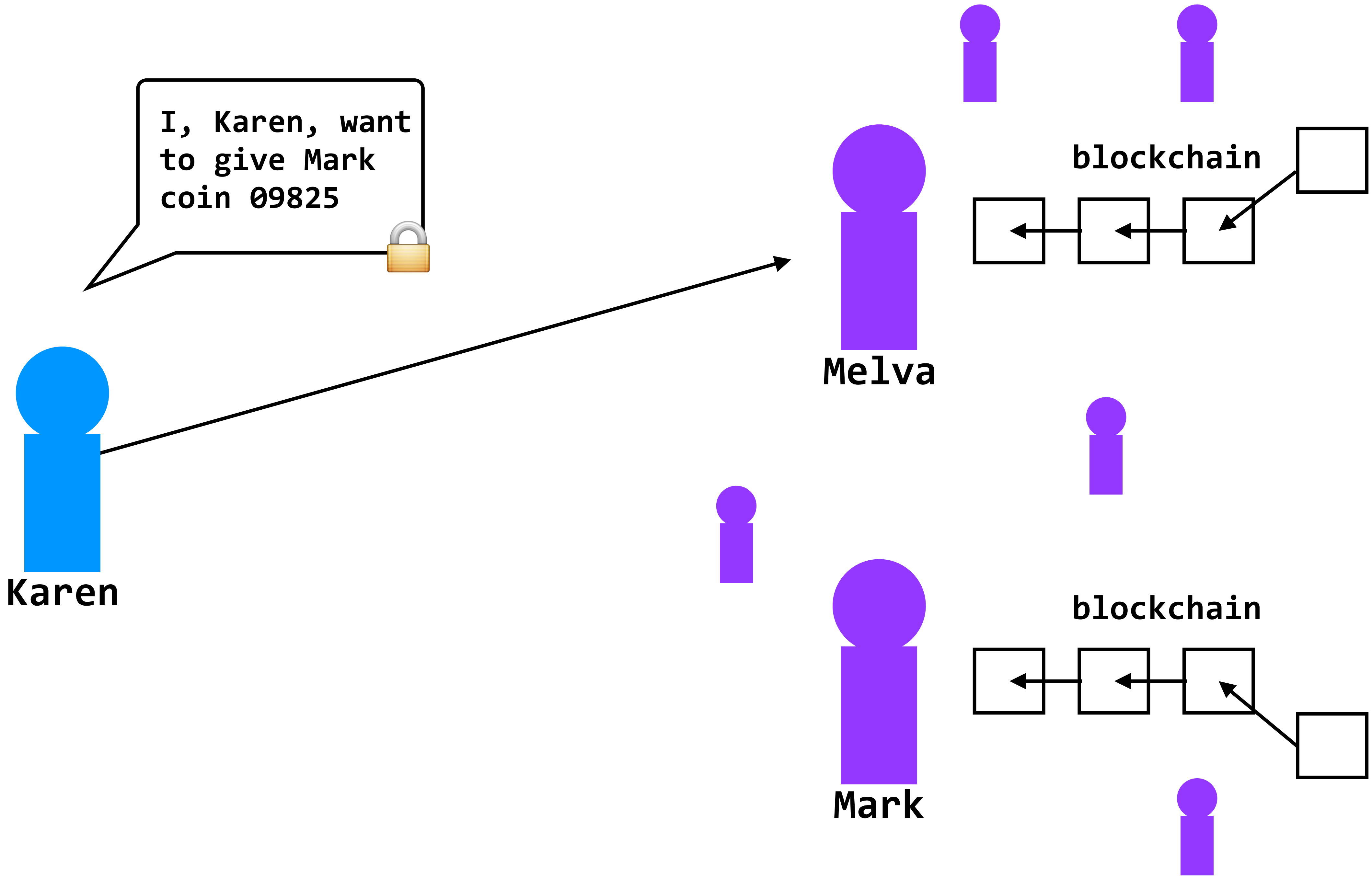


Melva uses her log to verify that Karen owns coin `09825`, and then sets about solving a **proof-of-work** to validate this block of transactions

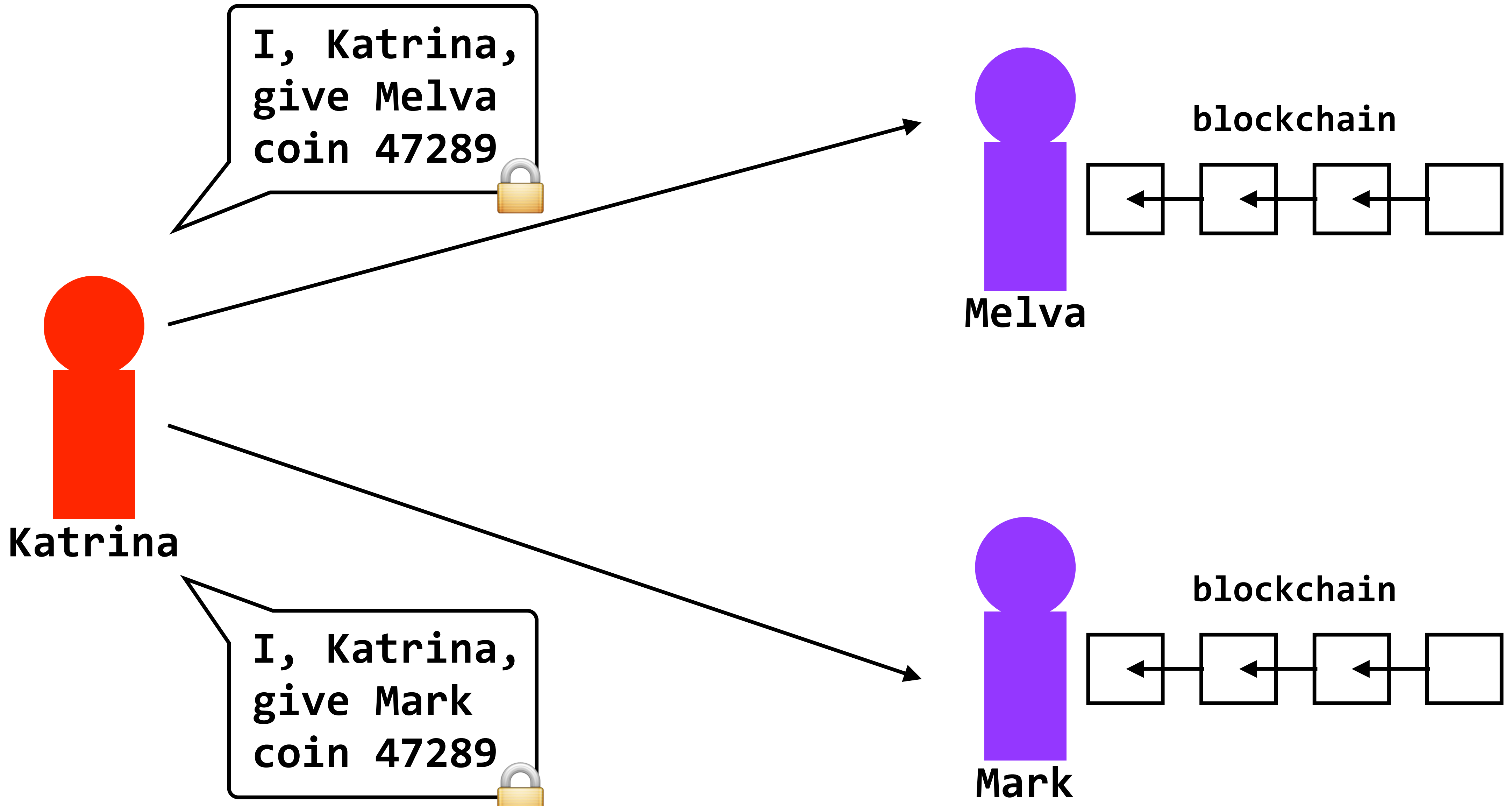
Once she solves it, Melva broadcasts the block along with the solution to the rest of the network, and gets a monetary reward



If multiple users “win” the competition at (roughly) the same time, the blockchain will **fork**

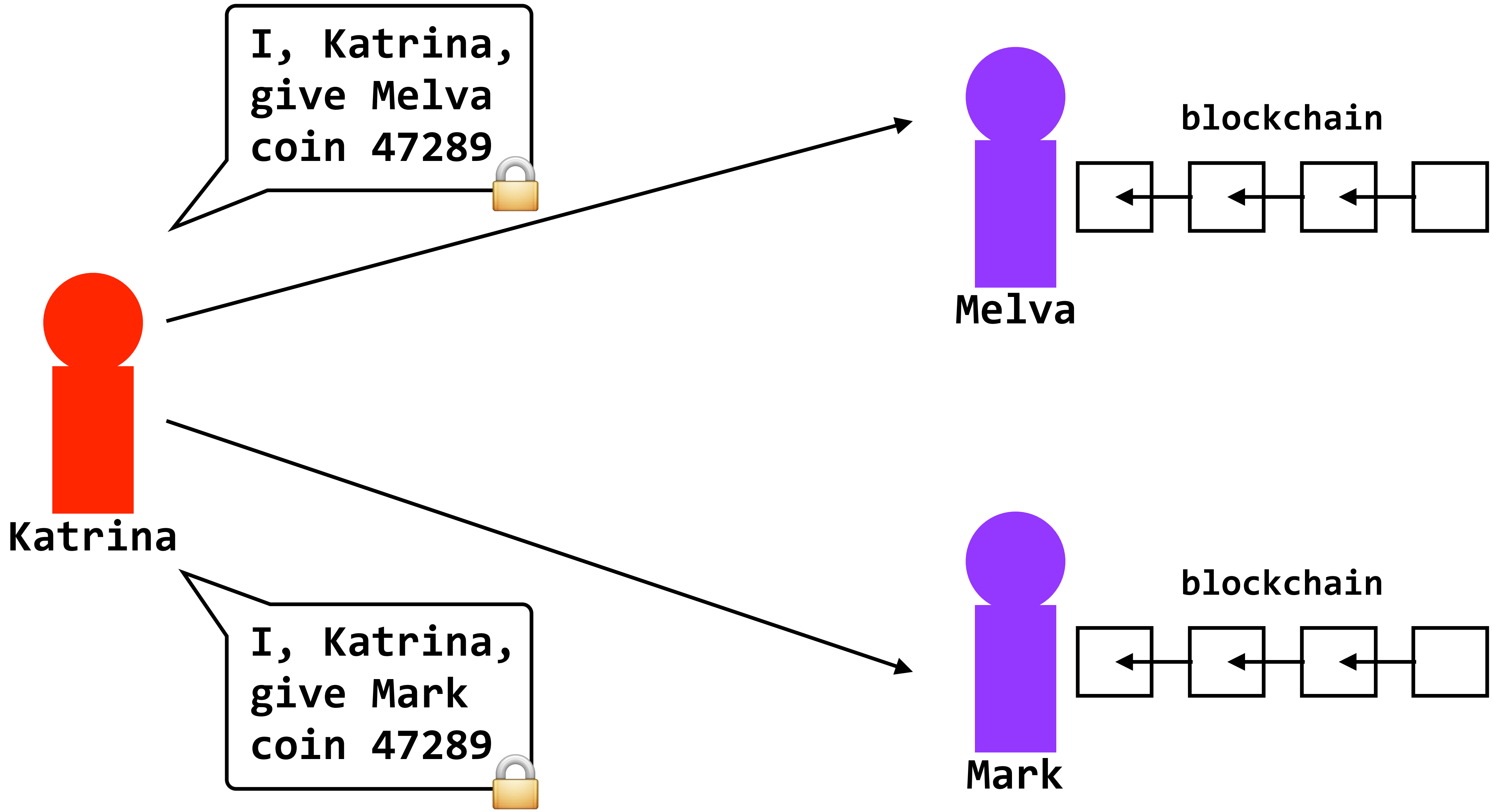


If multiple users "win" the competition at (roughly) the same time, the blockchain will **fork**. Bitcoin resolves this problem by having miners work only on the longest fork, quickly rendering the other branch obsolete



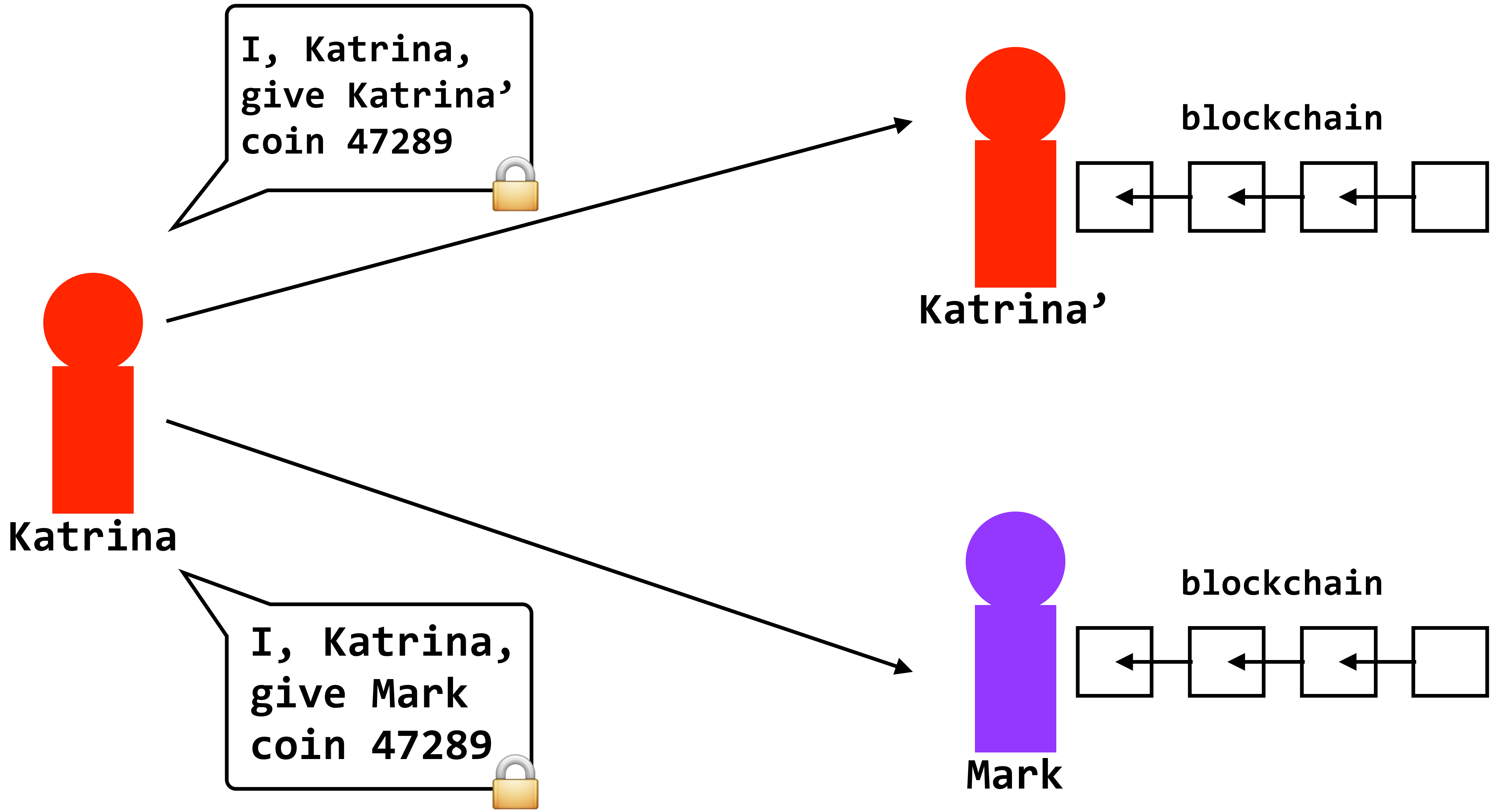
idea: Katrina tries to validate a block that includes both of those transactions

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 Mark 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.