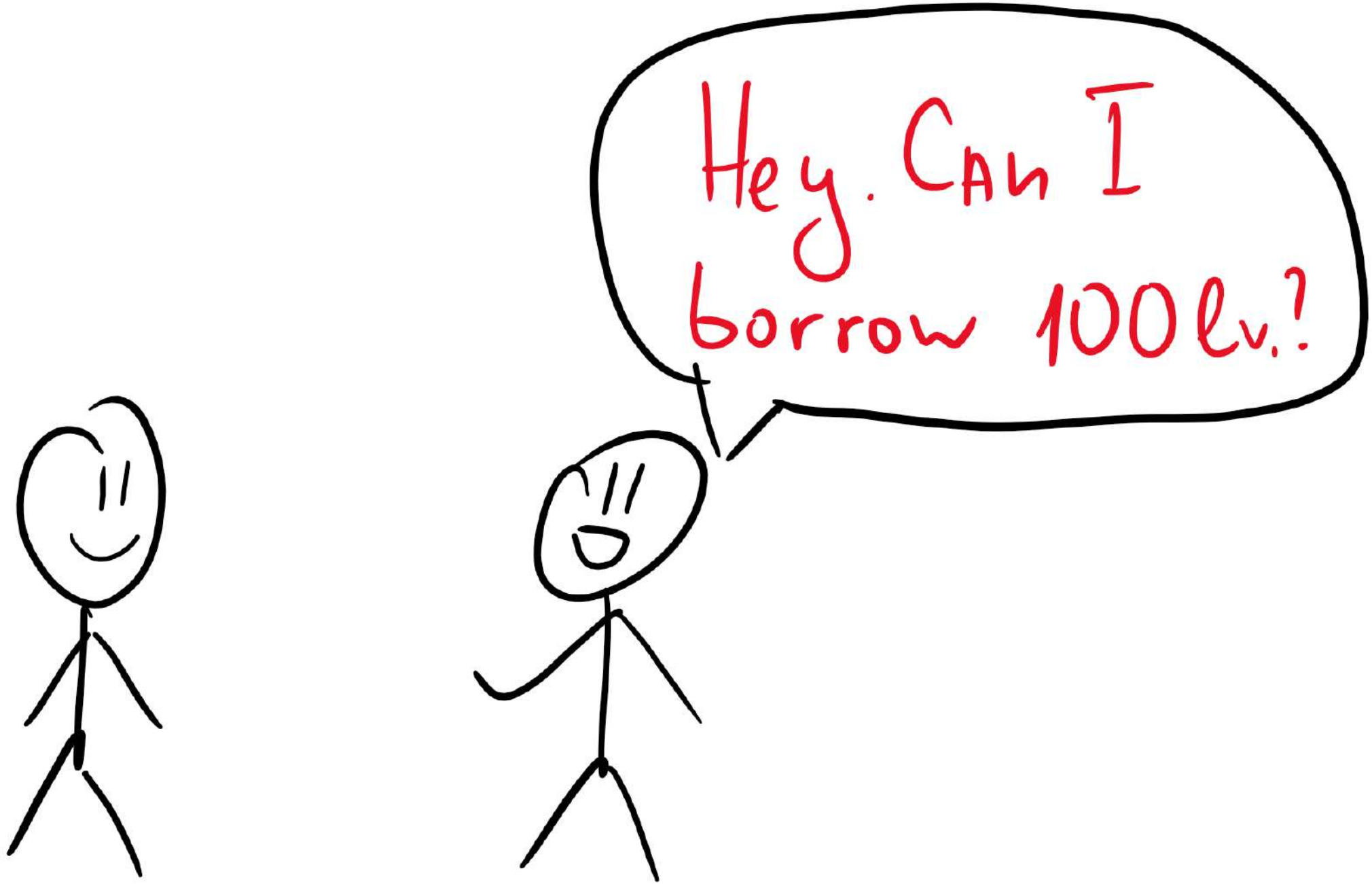
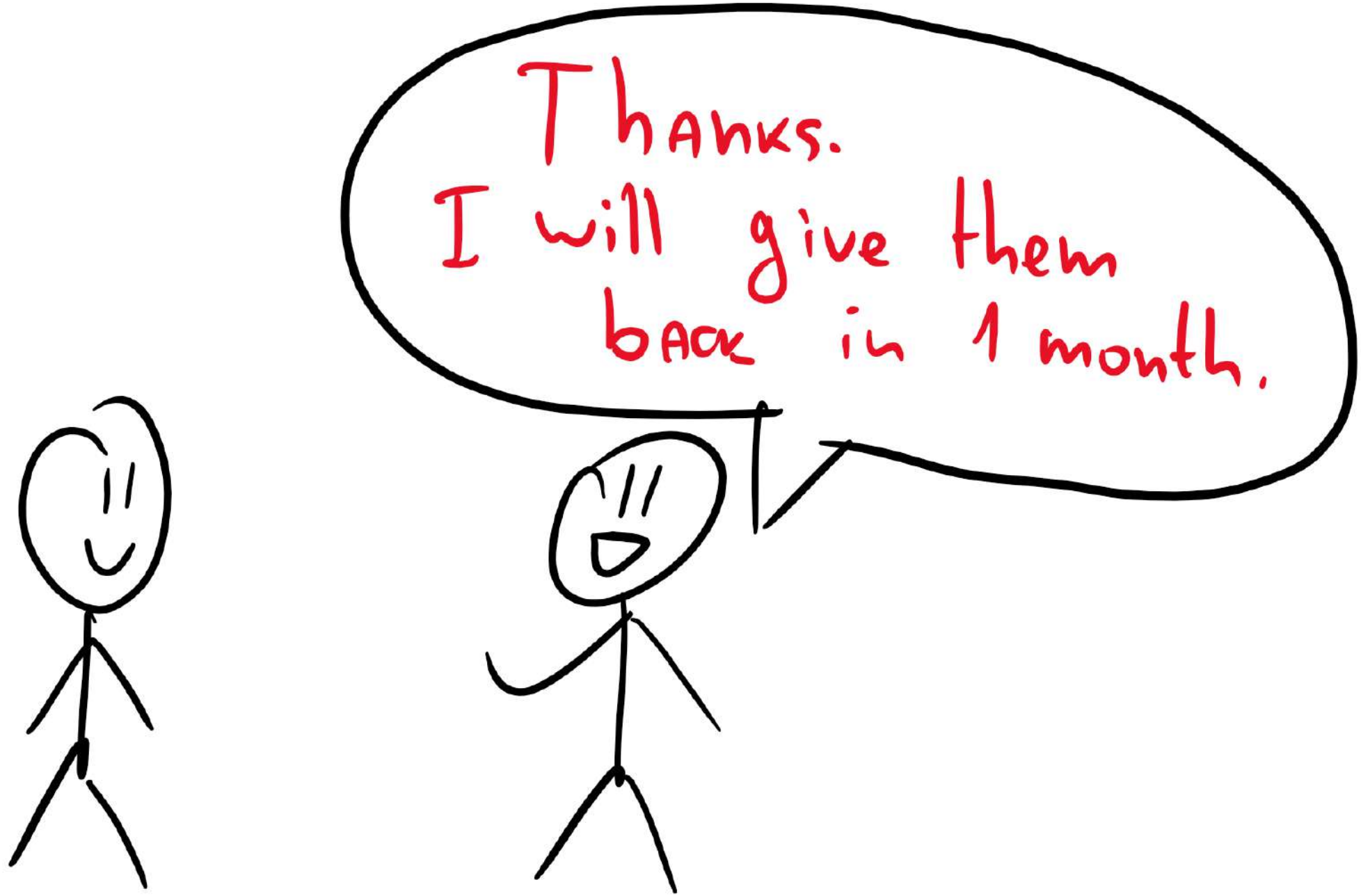

Hey. Can I
borrow 100 lv.?


Sure. Here you go!


Thanks.
I will give them
back in 1 month.

One month later...


Hey!
Where's my money?


What money?

Trust


What money?
КТБ
Where's our money?

Trust

Trustless

Learn Blockchain

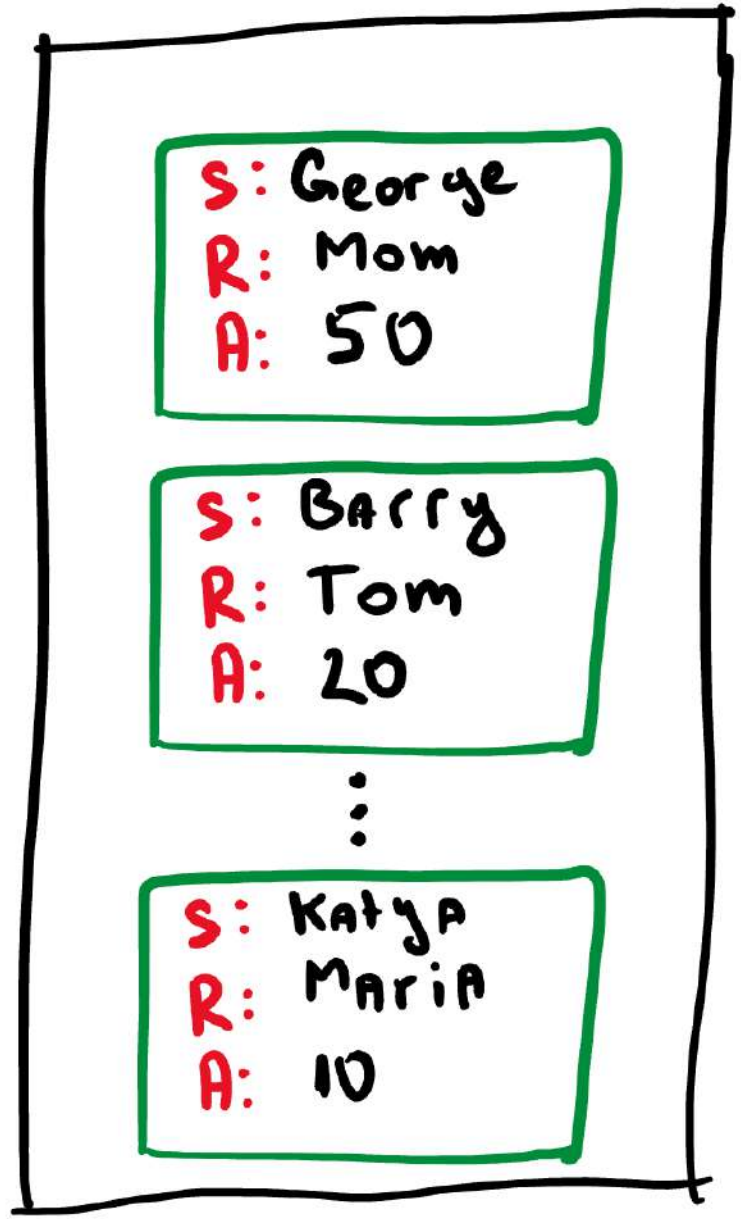
By Building It

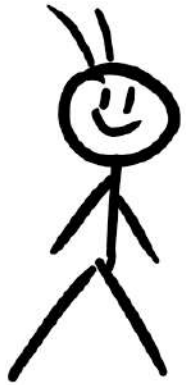
Preslav Mihaylov

/preslavmihaylov

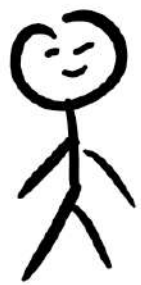
Part I:

Creating our own financial system

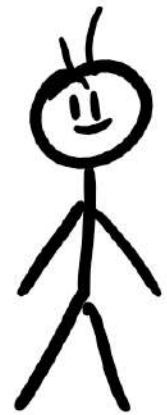

S: George
R: Mom
A: 50
S: Barry
R: Tom
A: 20
S: Katya
R: Maria
A: 10



You

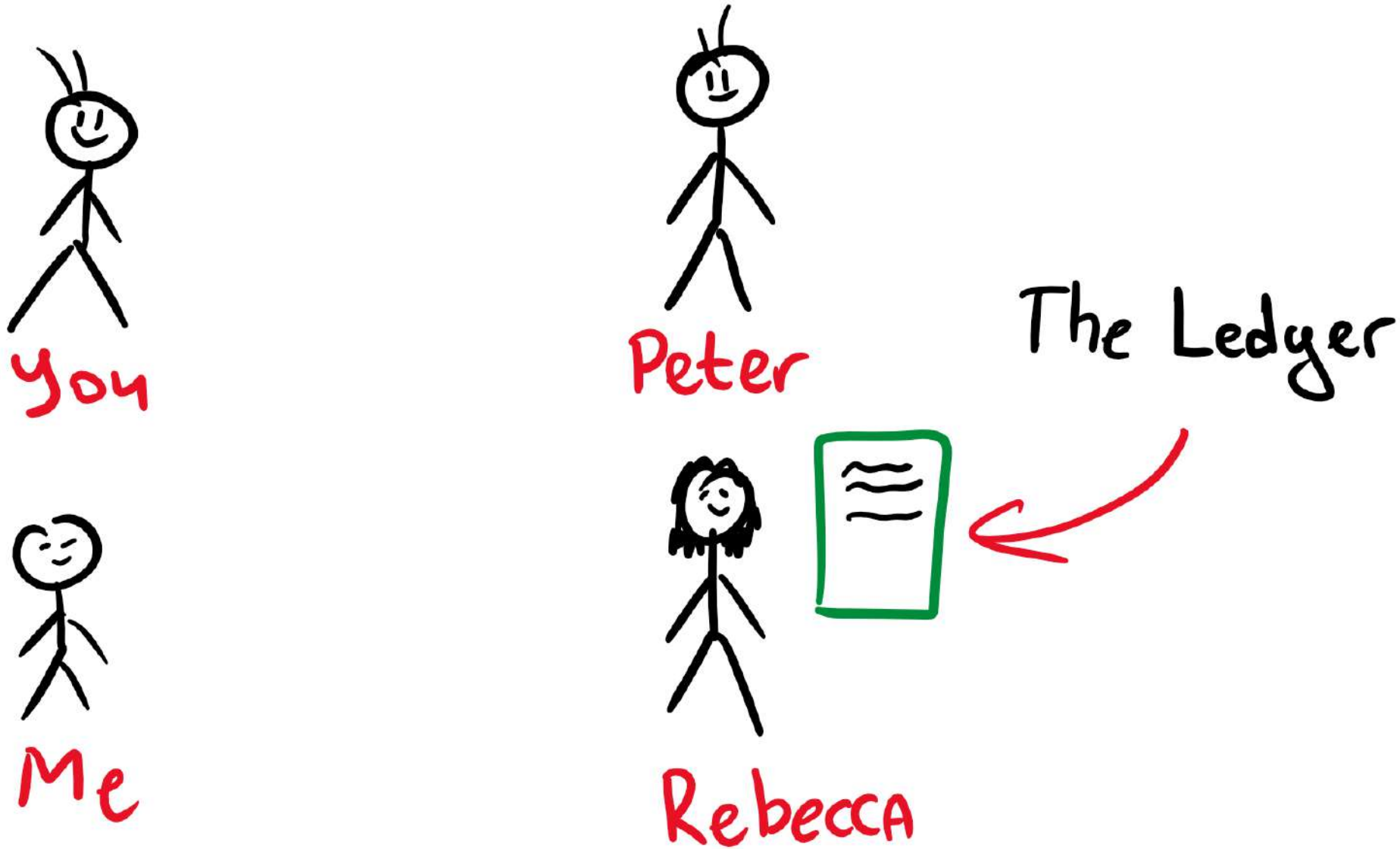


Me

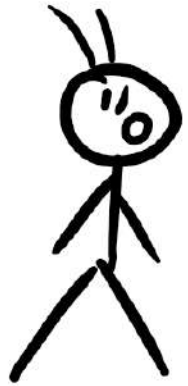


Peter

Rebecca


You
Me
Peter
The Ledger
Rebecca

Security

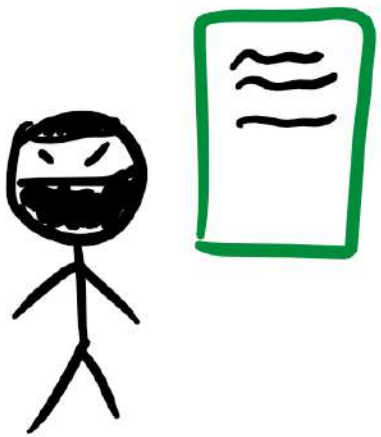


You

Me

Peter

Rebecca

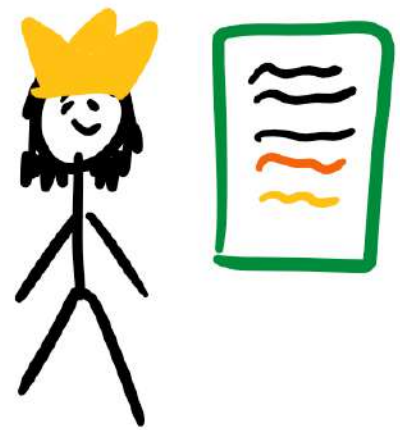


Trust

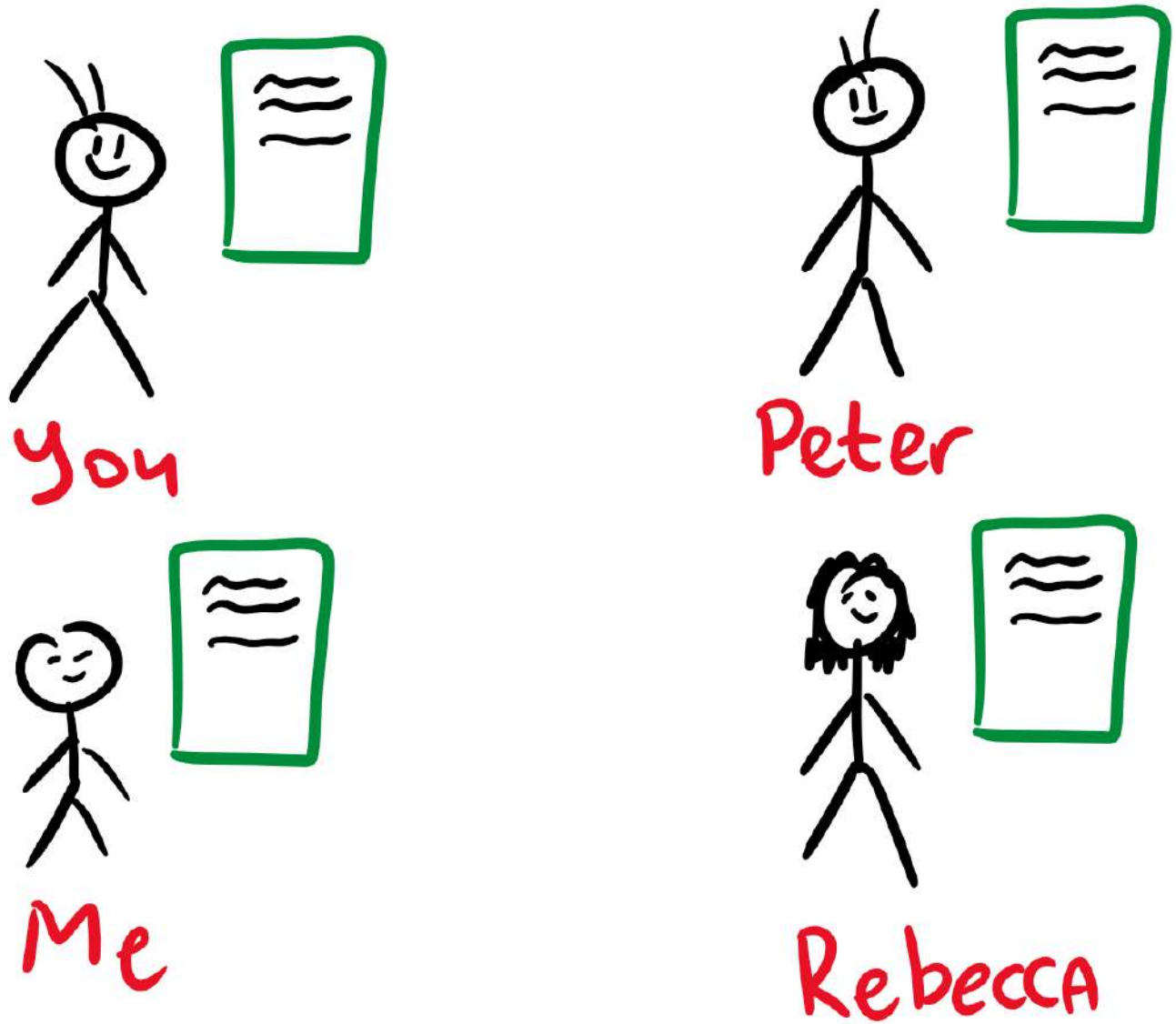
You

Me

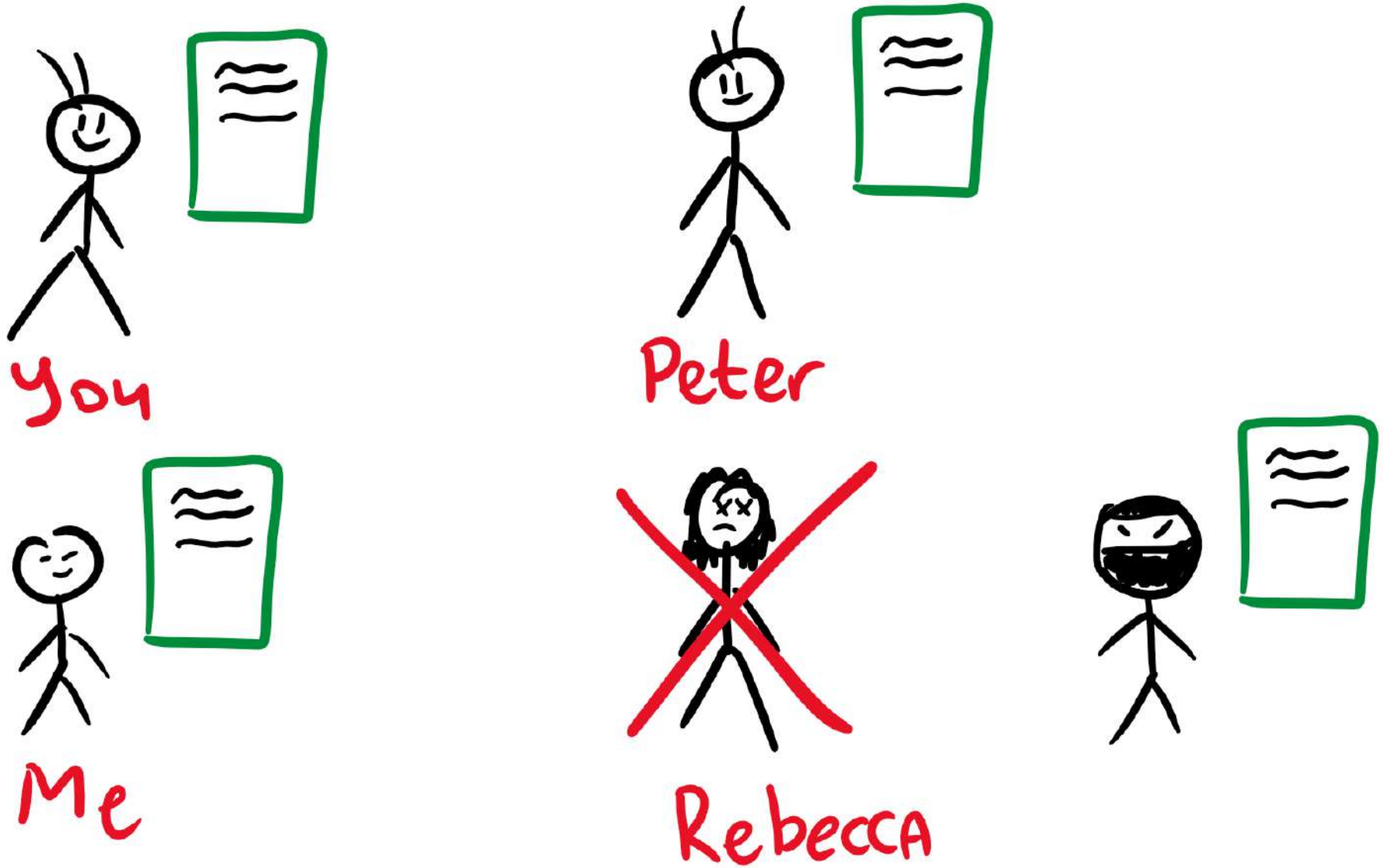
Peter



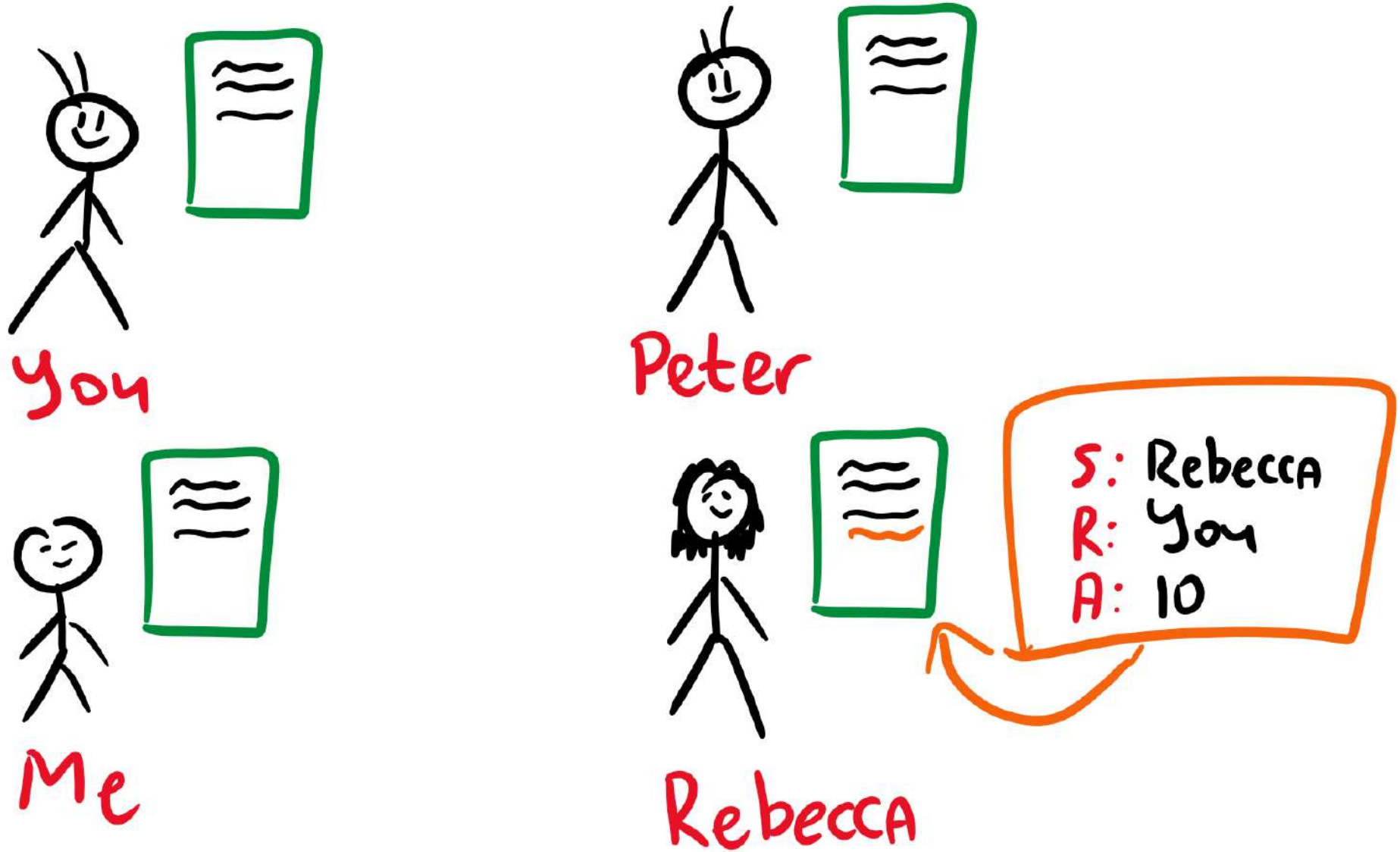
Rebecca

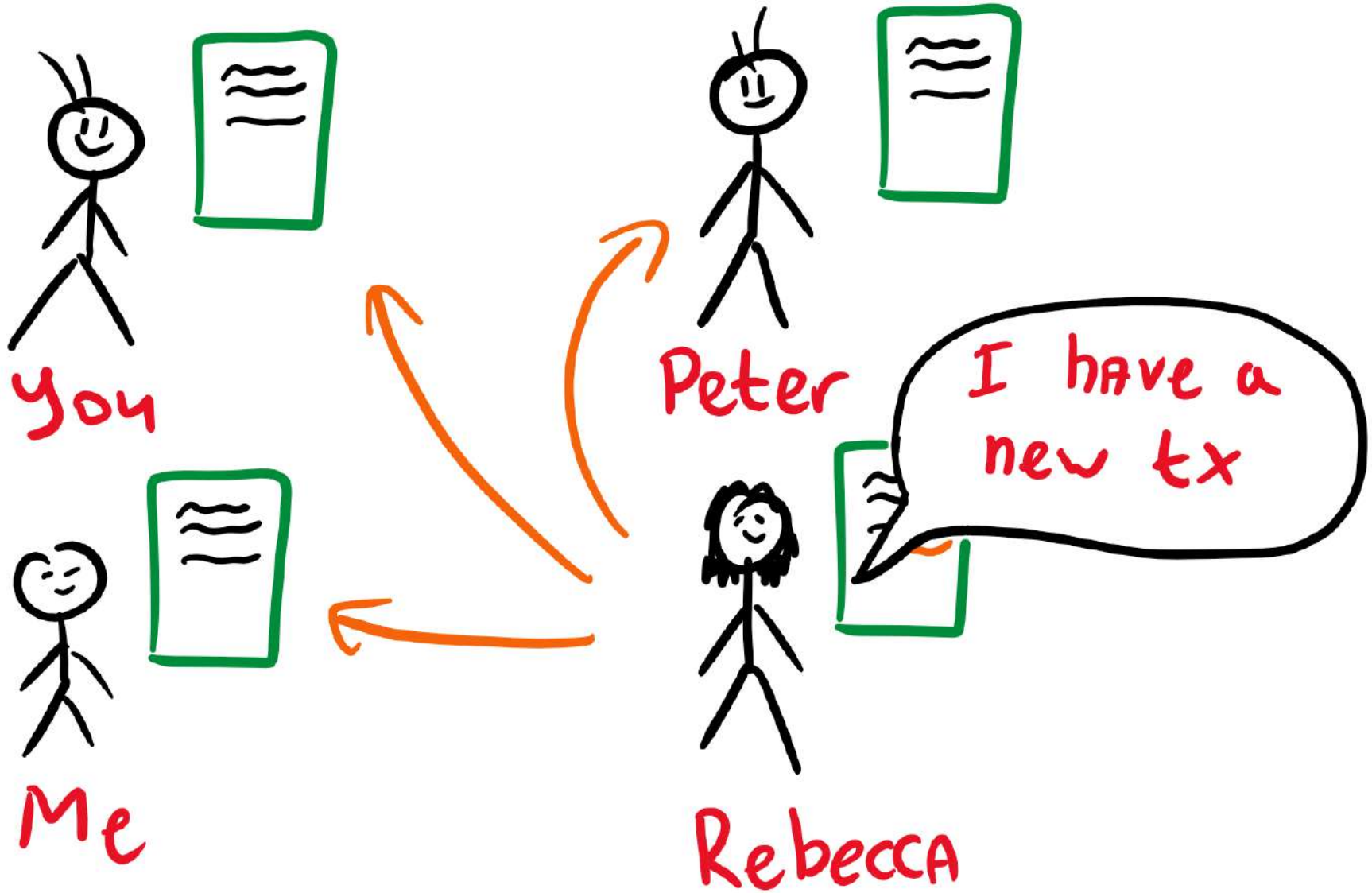

You
Peter
Me
Rebecca

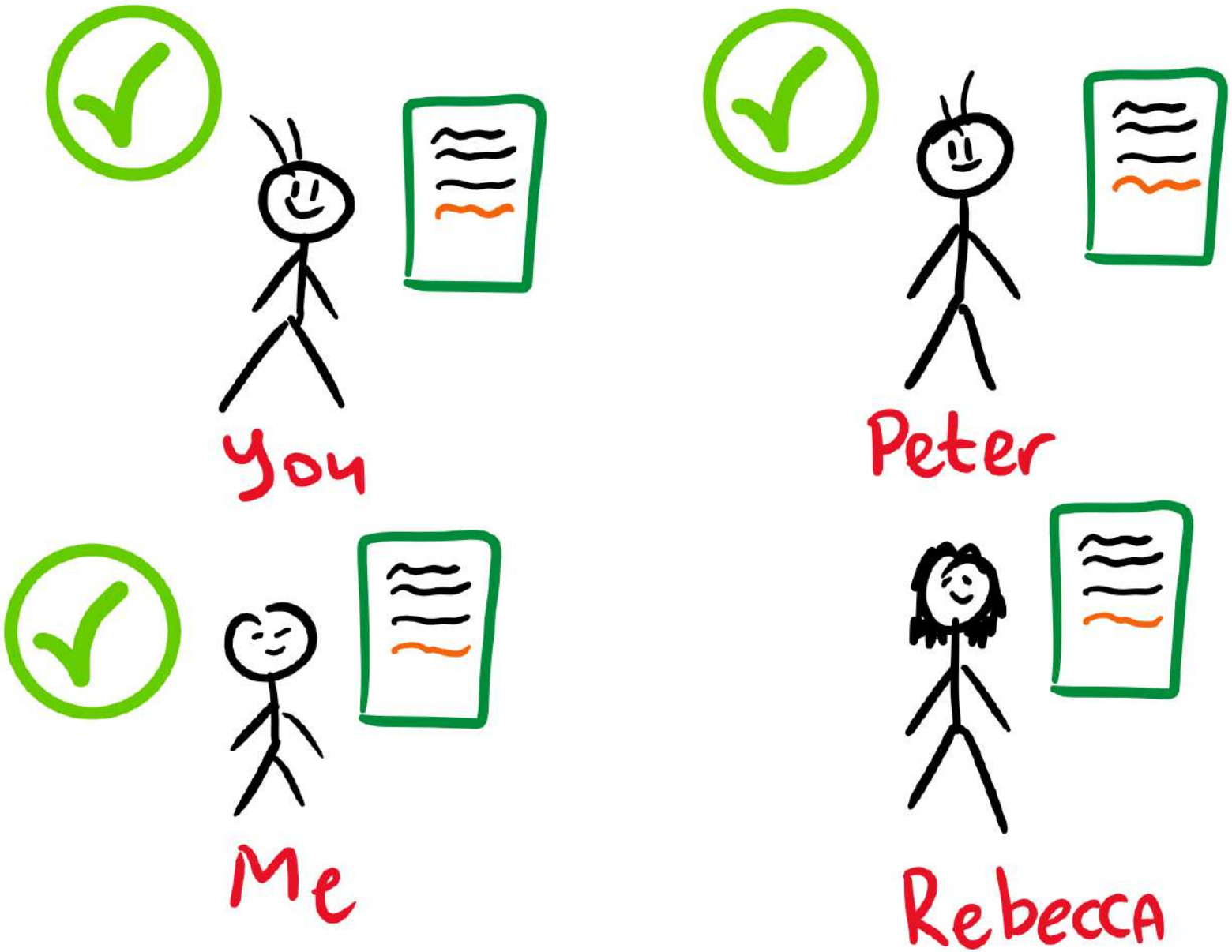
Security

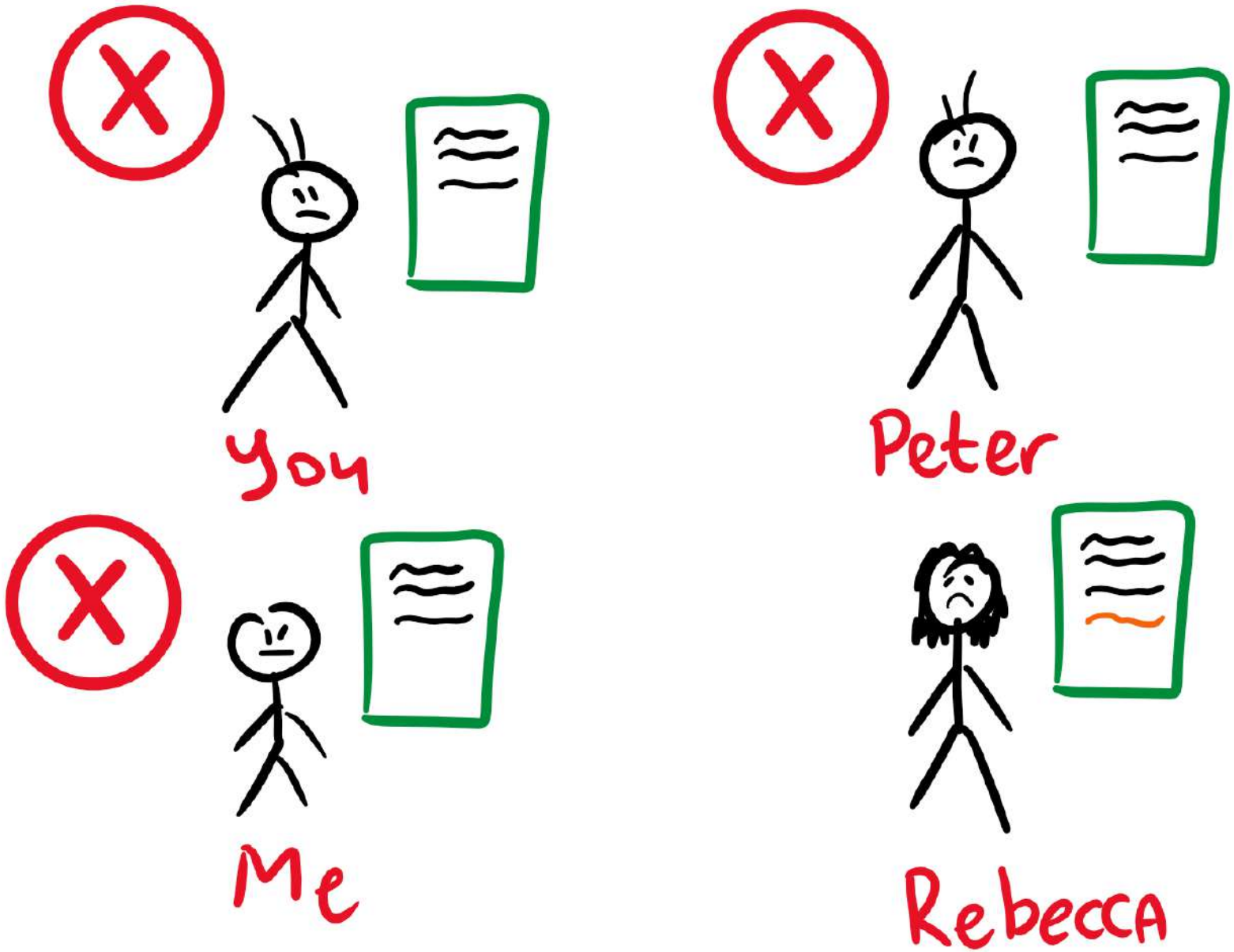

You
Peter
Me
Rebecca

Trust


You
Me
Peter
Rebecca
S: Rebecca
R: You
A: 10


You
Peter
I have a new tx
Me
Rebecca

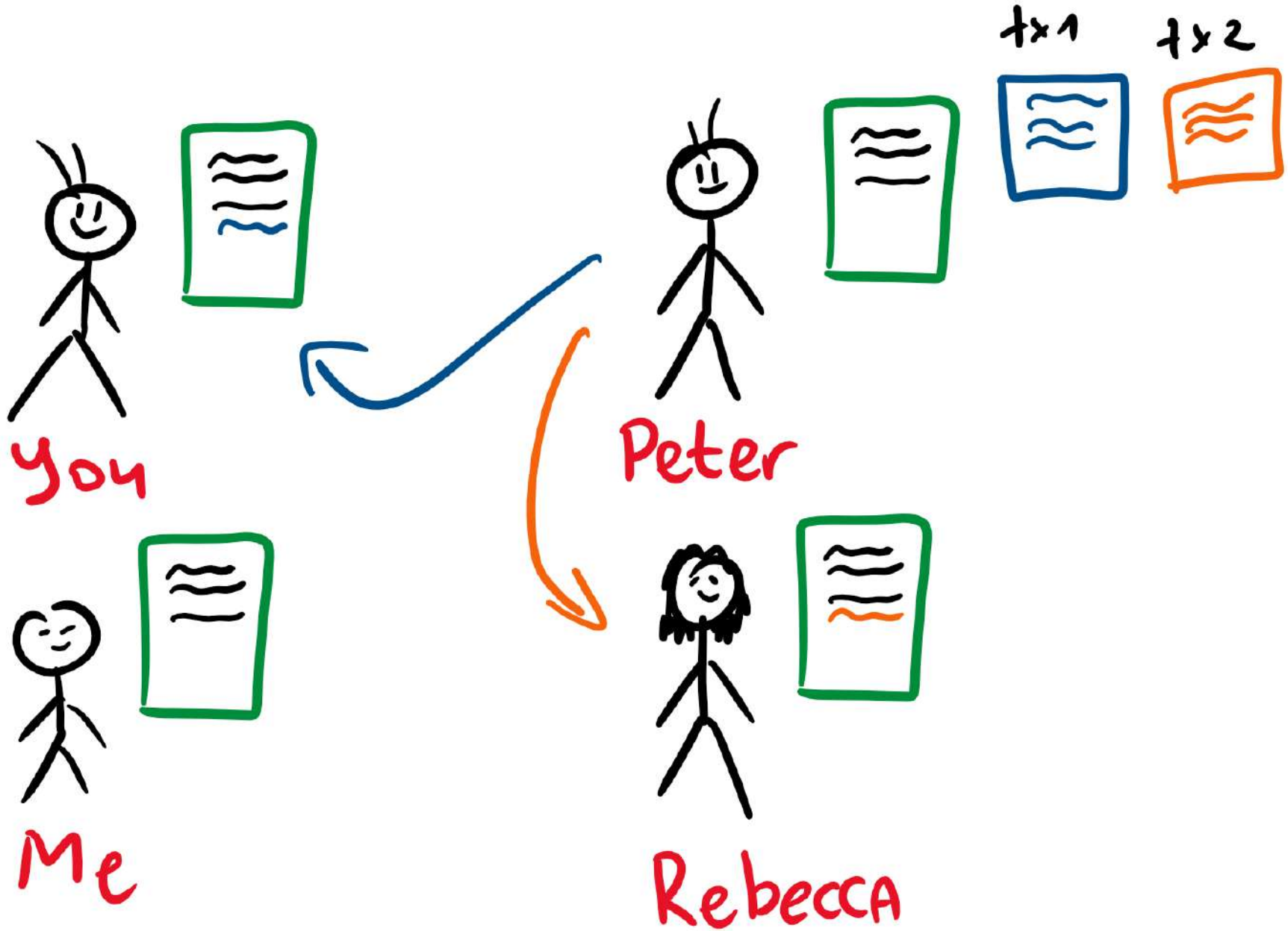

You
Peter
Me
Rebecca

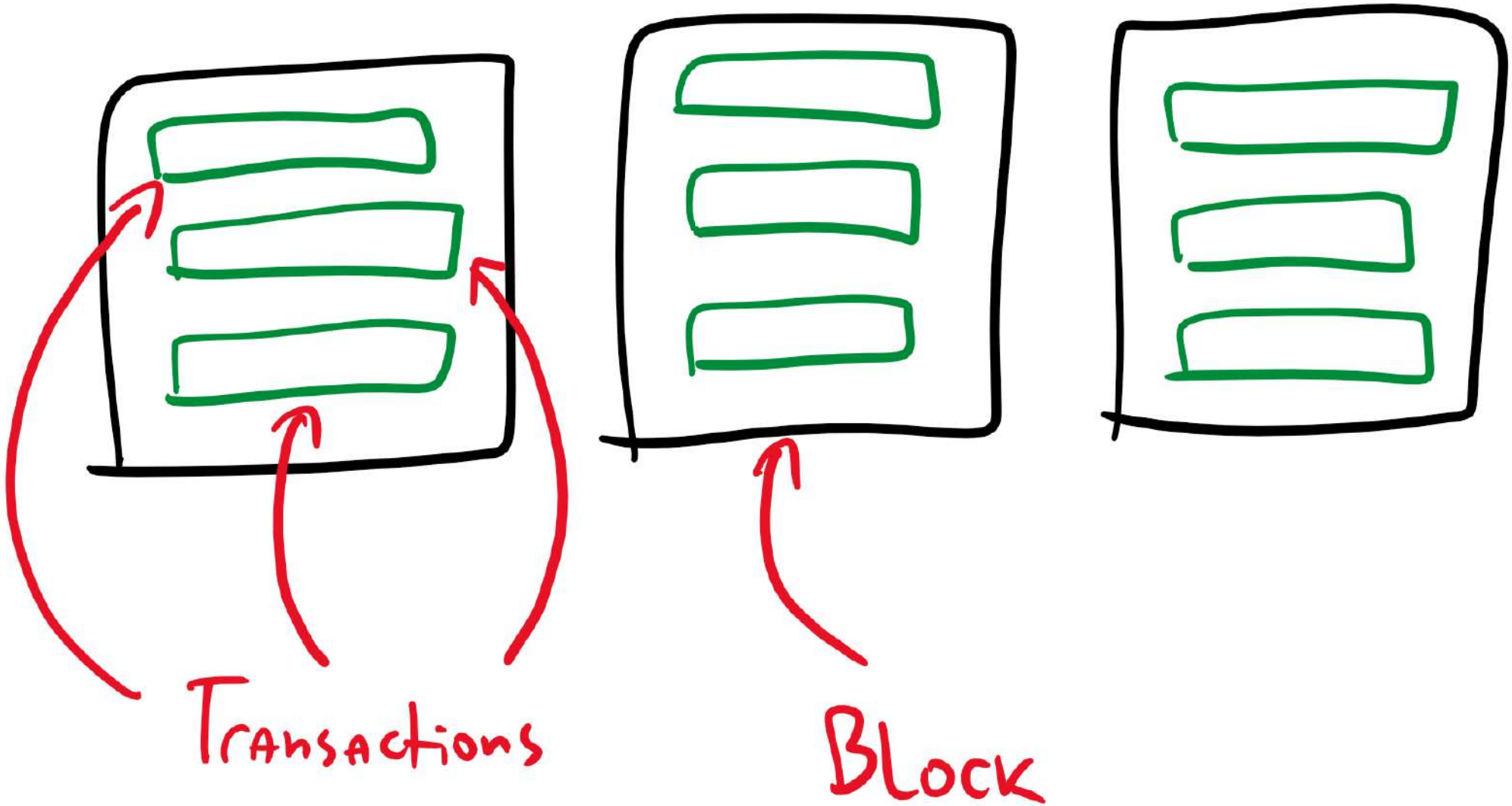

You
Peter
Me
Rebecca

Part II:

Building The Blockchain

Transaction Collisions

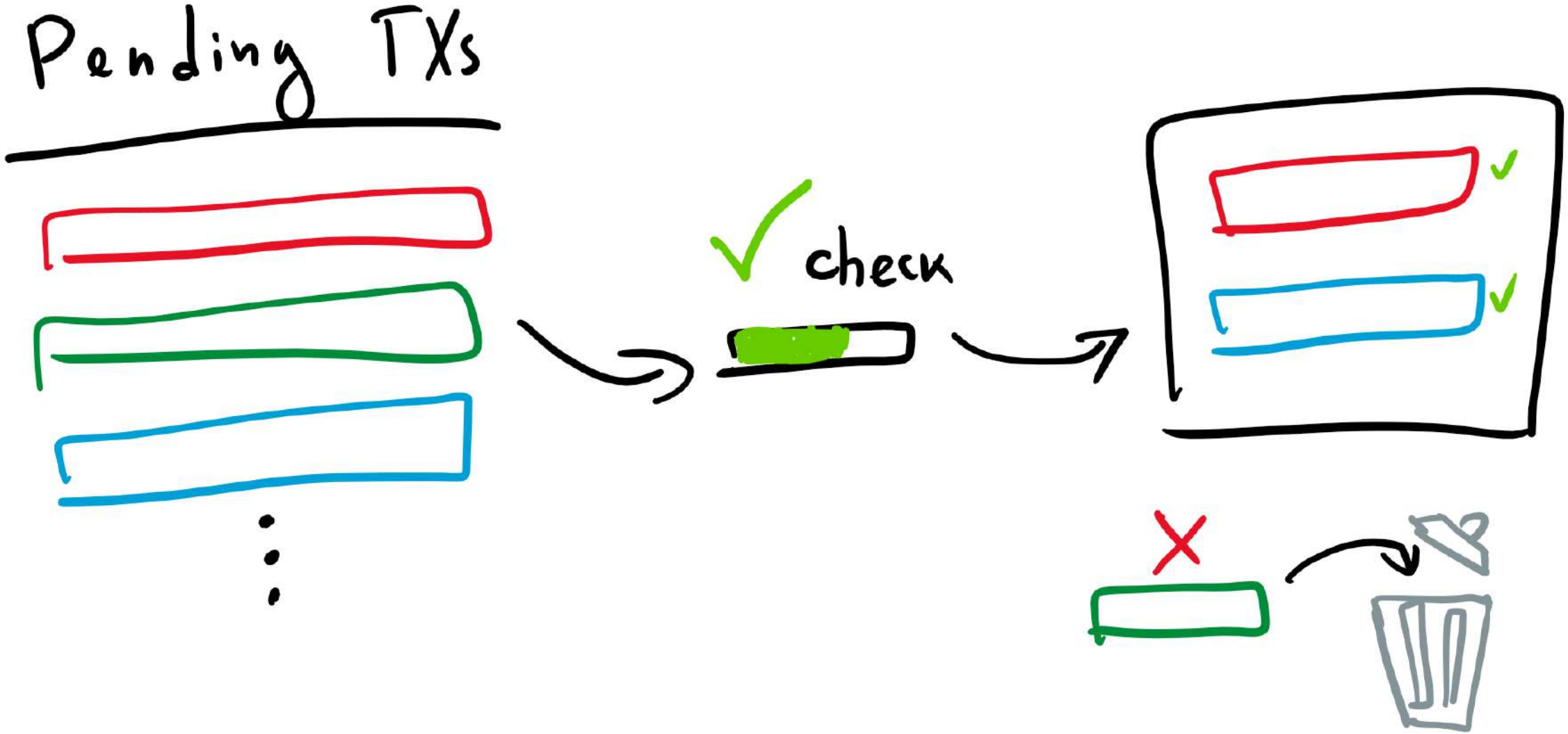

tx1
tx2
You
Peter
Me
Rebecca

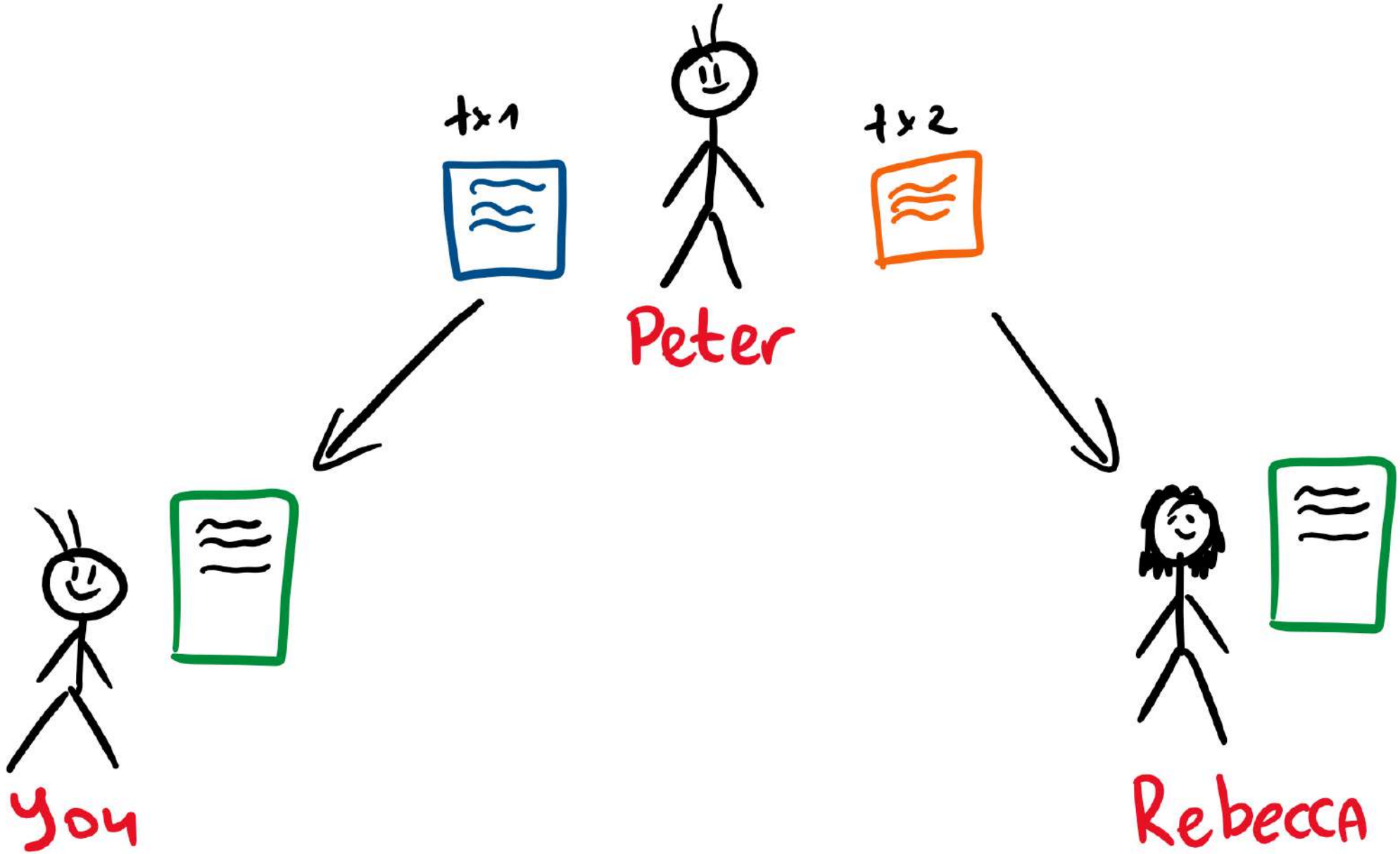

Transactions
Block

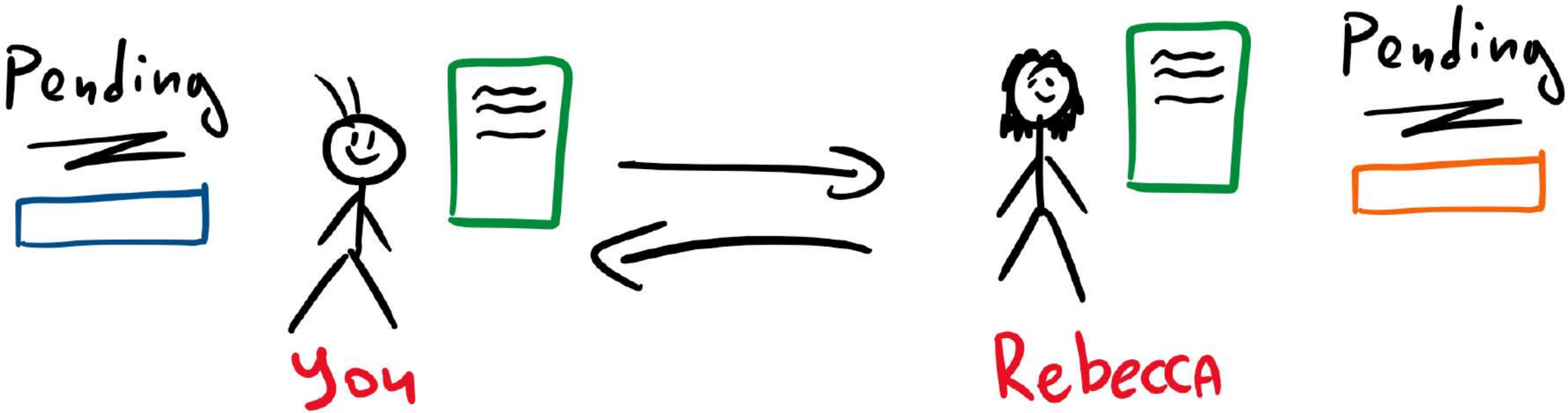
```java
public class Transaction {
	private String sender;
	private String recipient;
	private int amount;

	// constructor, getters, setters, toString...
}
```

```
public class Block {
      private final List<Transaction> transactions;

      // constructor, getters, setters, toString...
}
```

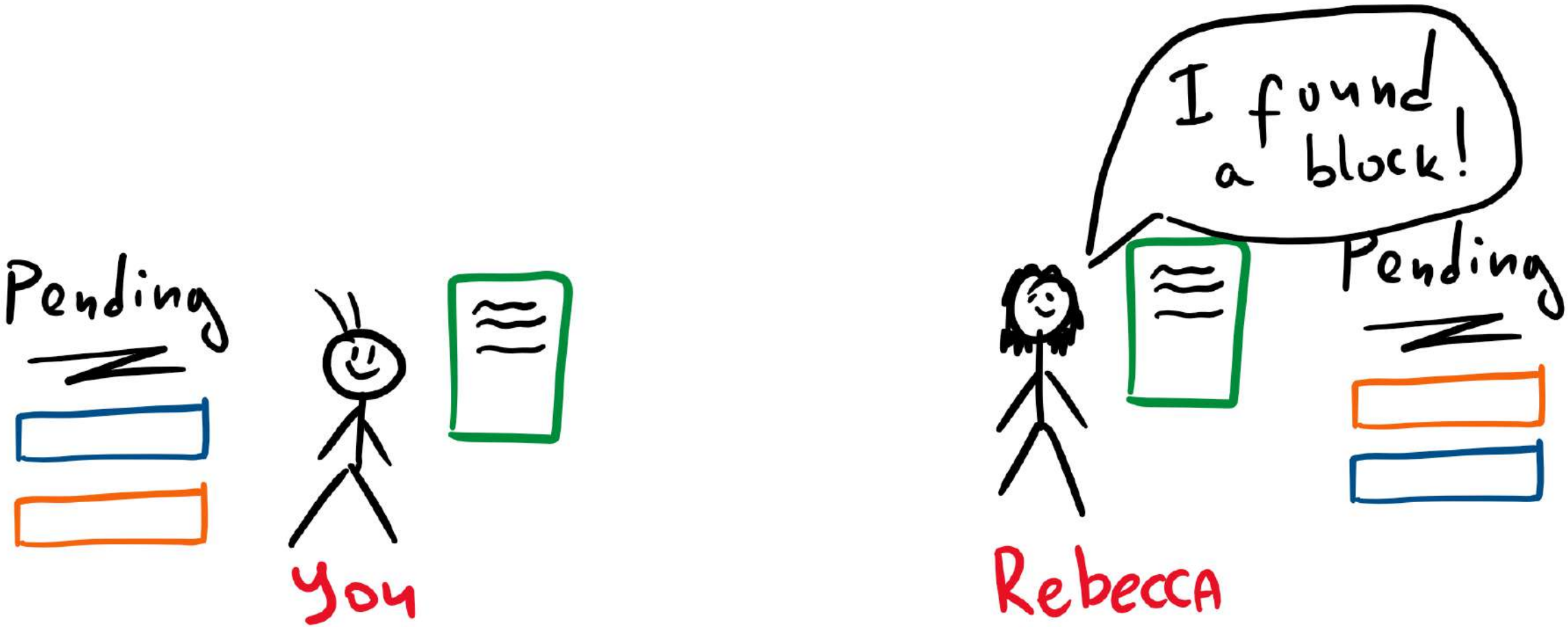
```
public class Blockchain {
	private List<Block> blocks;

	// constructor, getters, setters, toString...
}
```

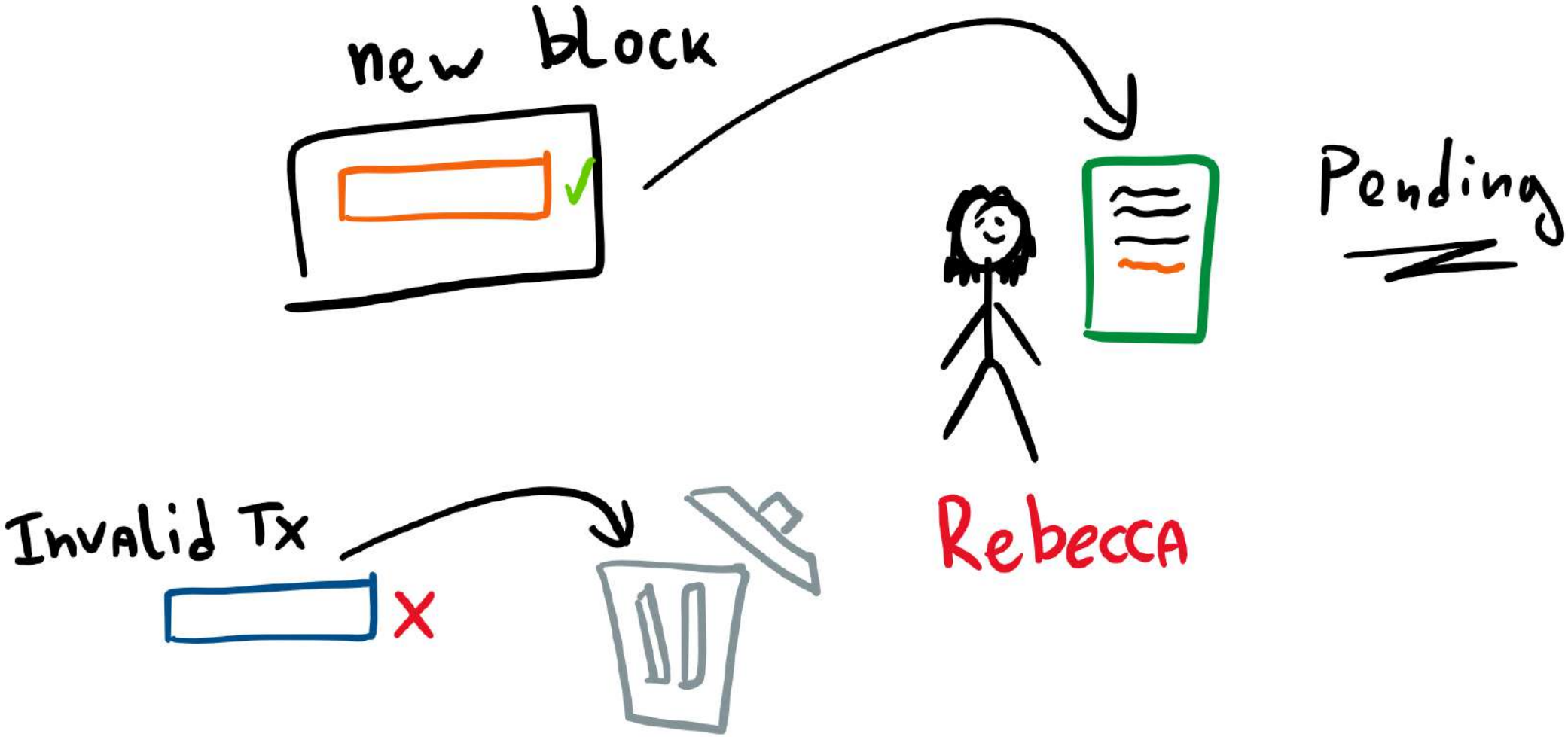

Pending TXs
check

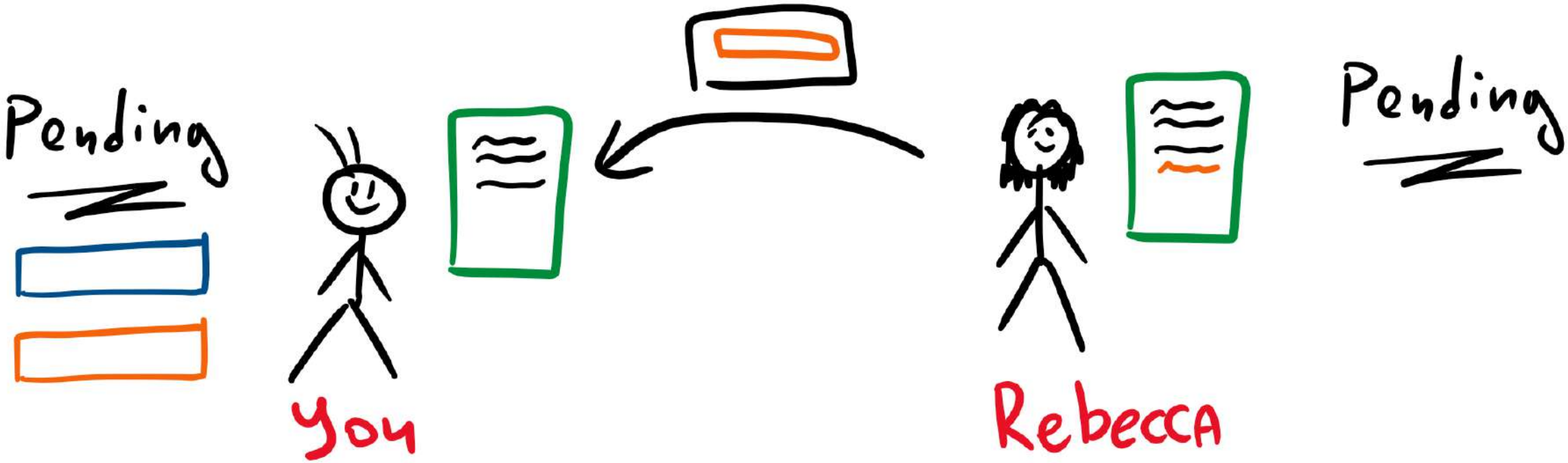

tx1
tx2
Peter
You
Rebecca

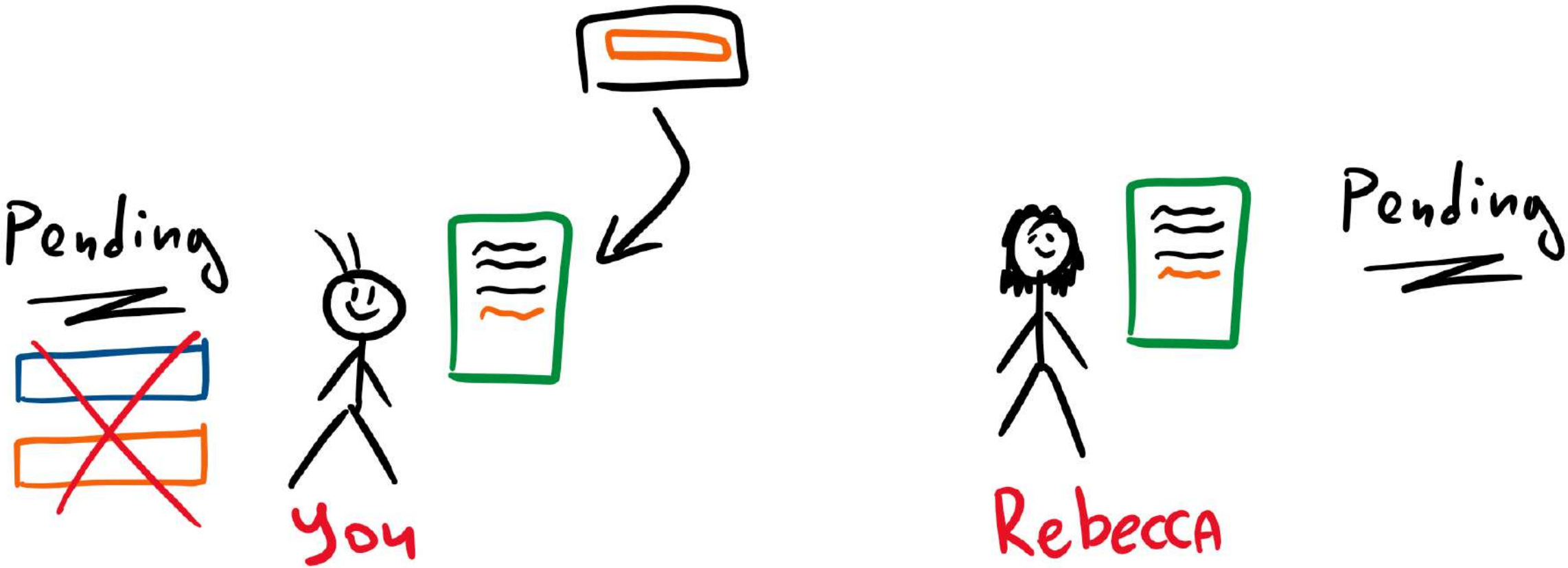

Pending
You
Rebecca
Pending


Pending
You
Rebecca
Pending


Pending
You
I found a block!
Pending
Rebecca


new block
Pending
Invalid Tx
Rebecca

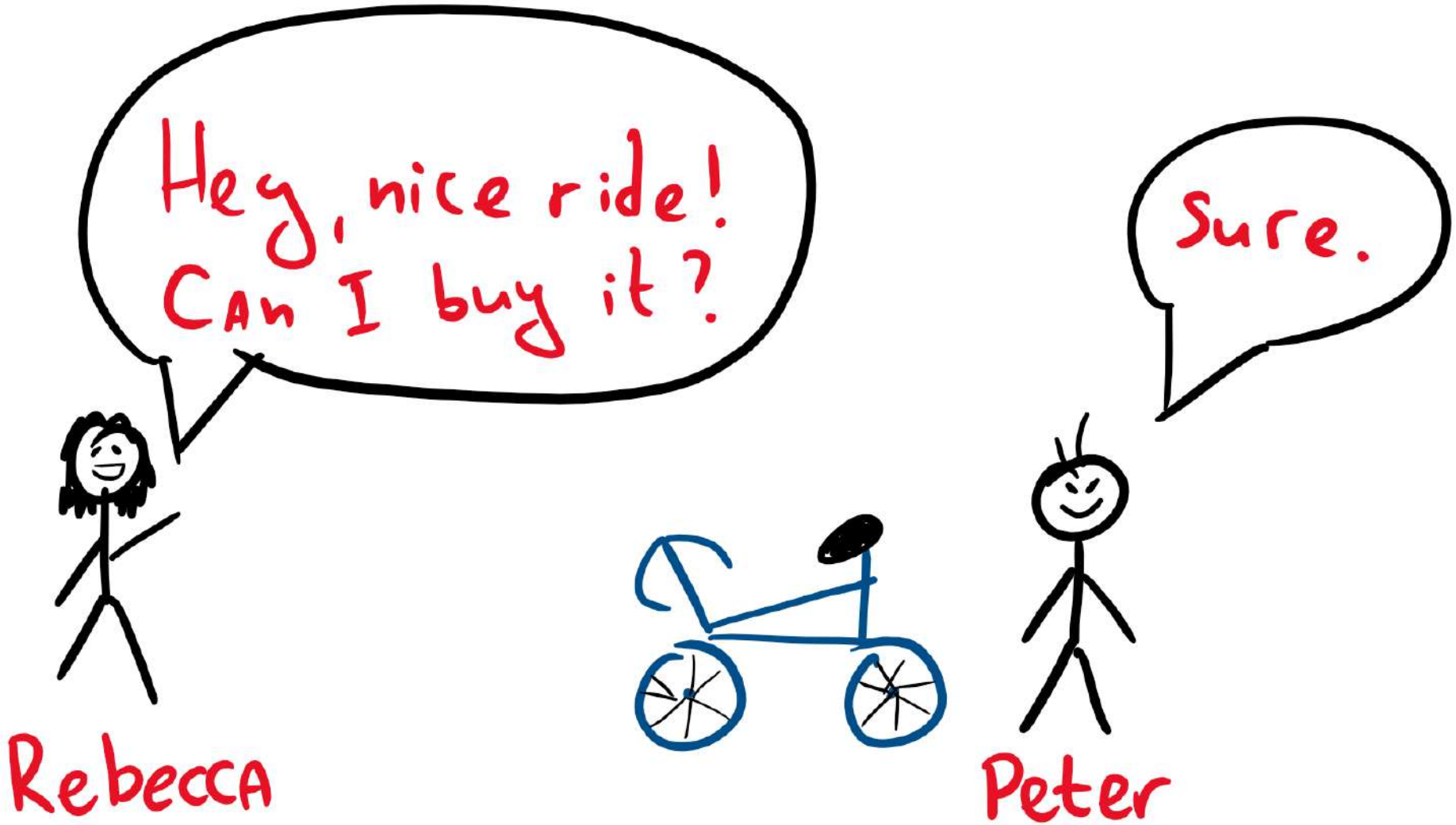

Pending
You
Rebecca
Pending

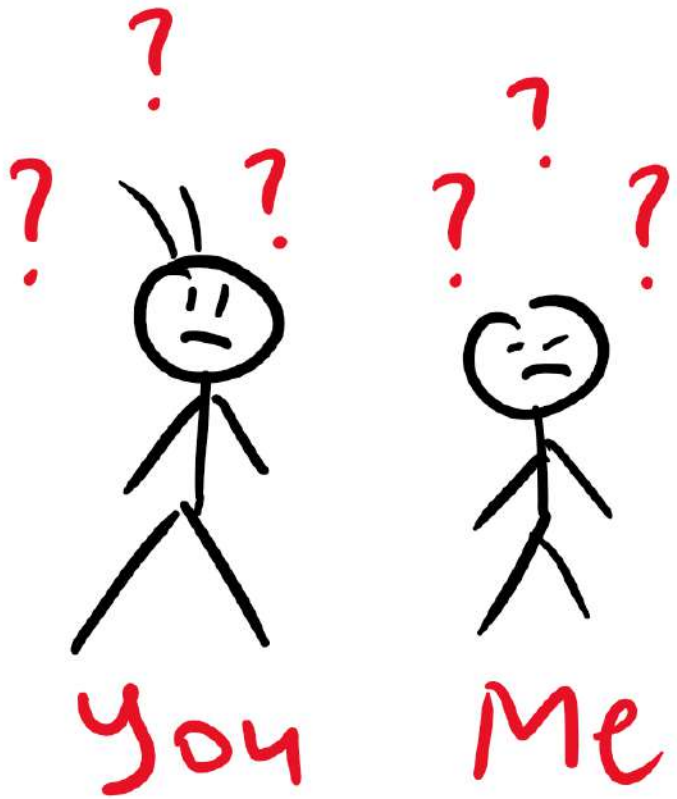

Pending
You
Rebecca
Pending

```
public class Blockchain {
	private List<Block> blocks;
	private List<Transaction> pendingTransactions;

	// constructor, getters, setters, toString...
}
```

```
// …
// TODO: Implement Peer Synchronization
// …
```

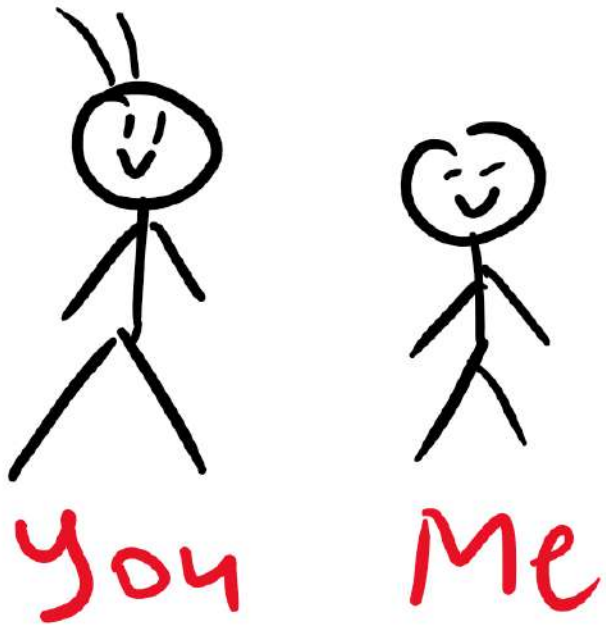
Transaction Validation

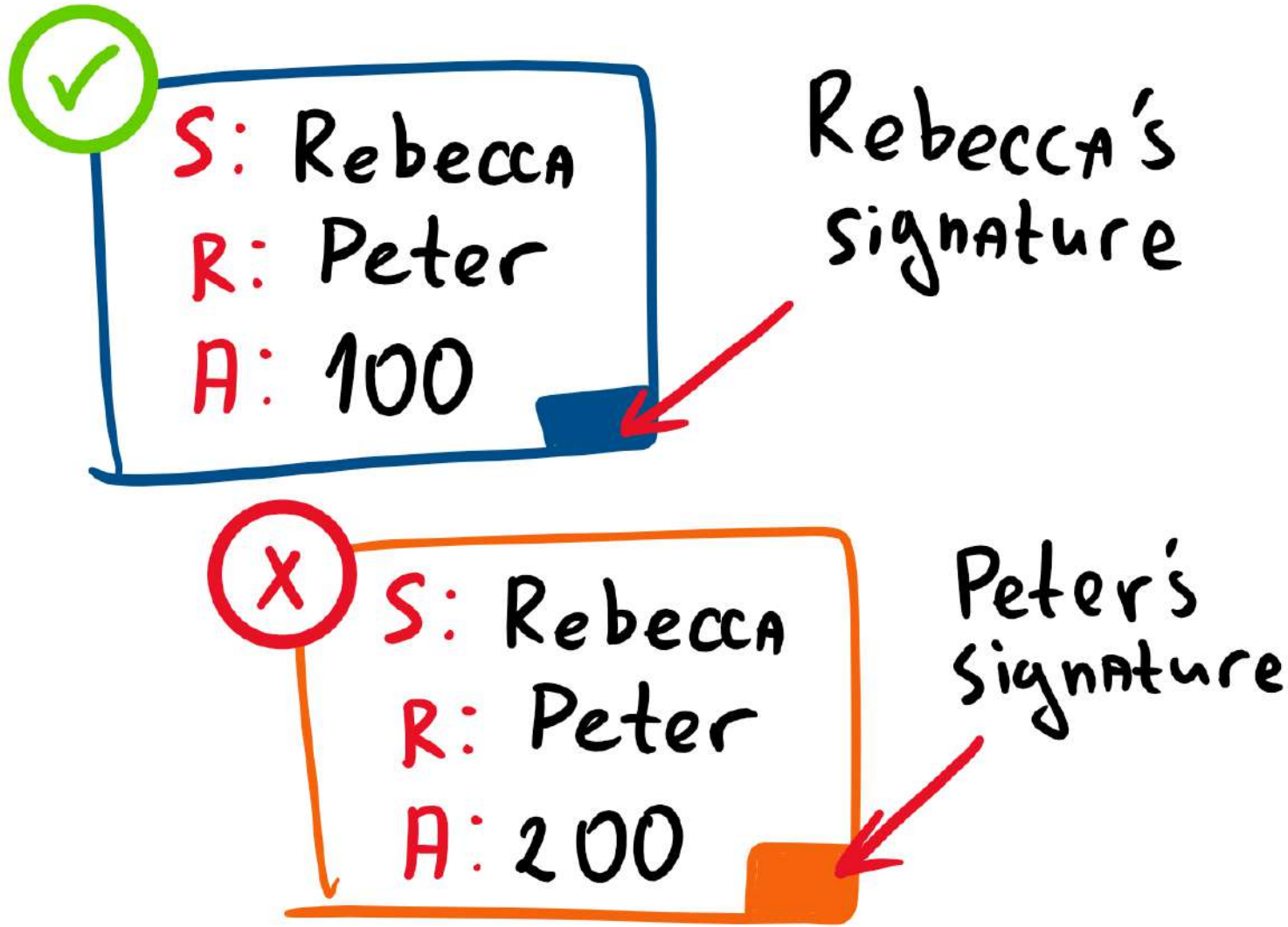

Hey, nice ride!
Can I buy it?
Sure.
Rebecca
Peter



S: Rebecca
R: Peter
A: 100

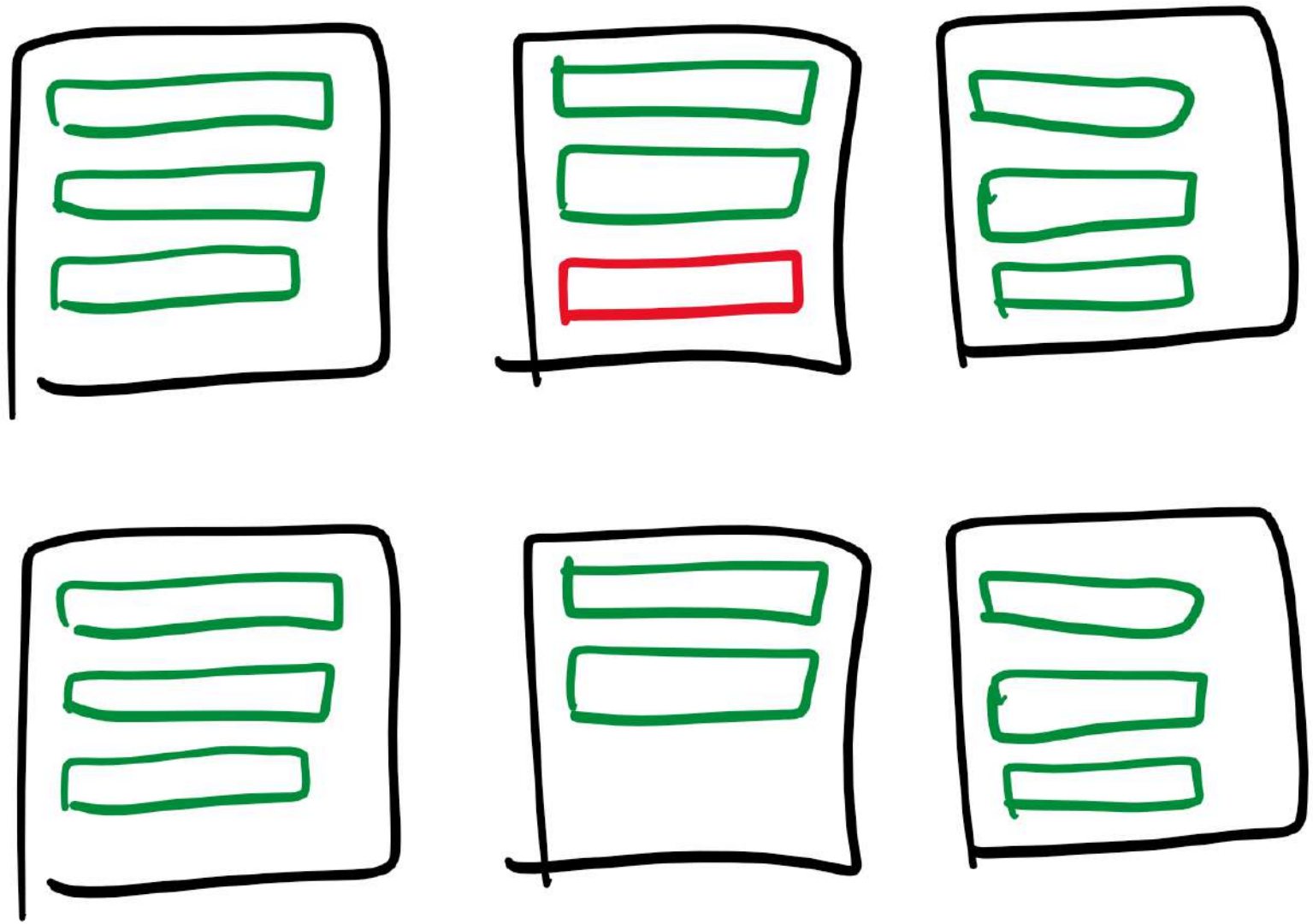
S: Rebecca
R: Peter
A: 200

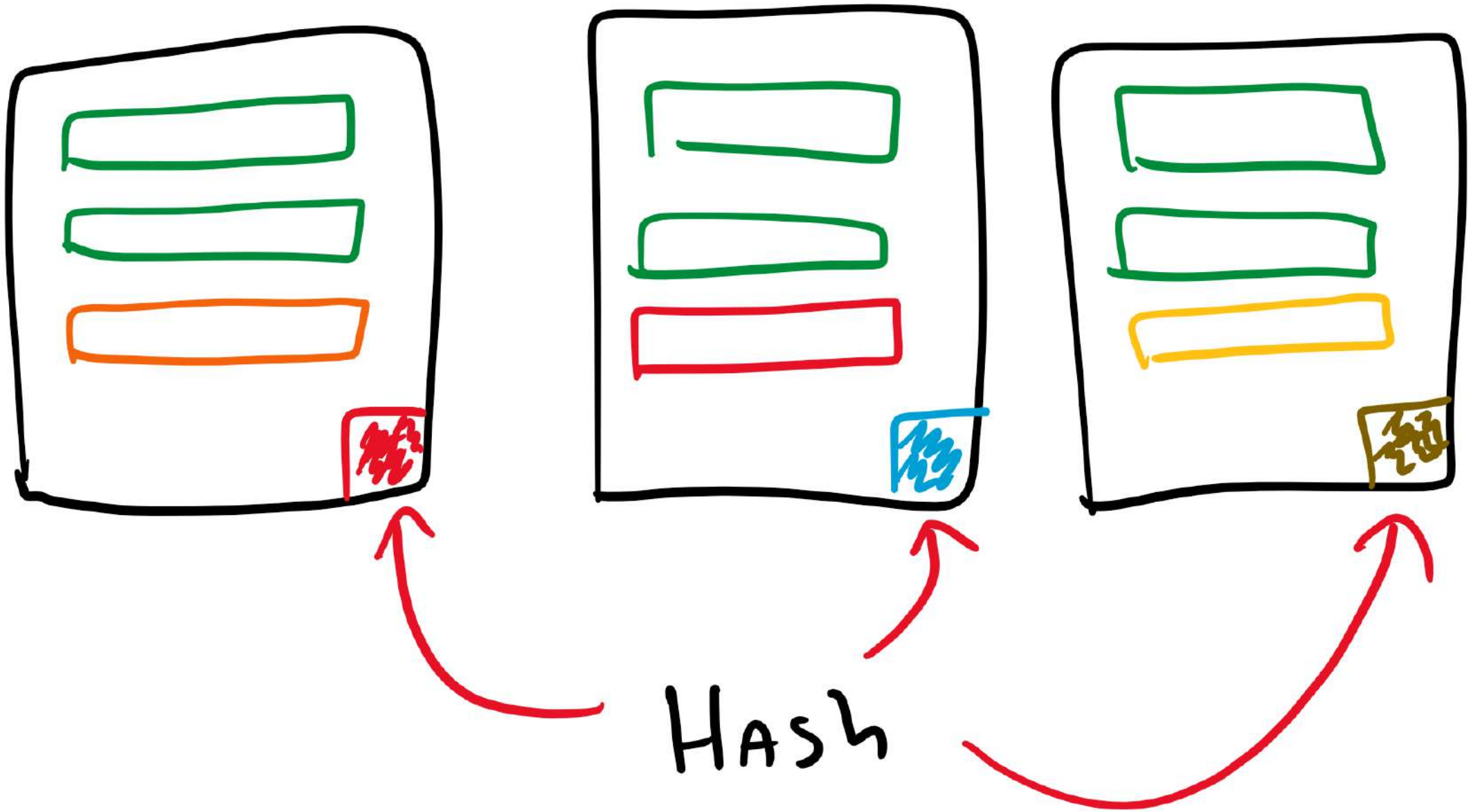

You
Me

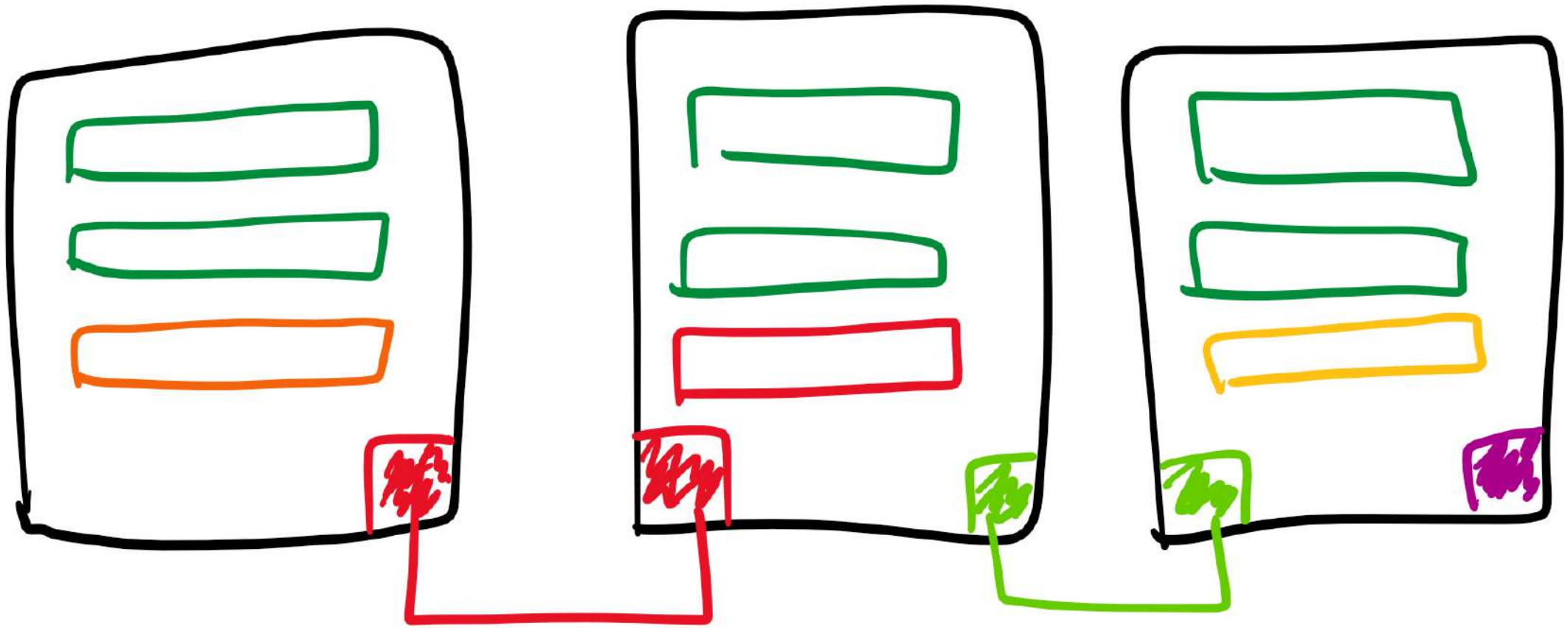

S: Rebecca
R: Peter
A: 100
Rebecca's signature
S: Rebecca
R: Peter
A: 200
Peter's signature

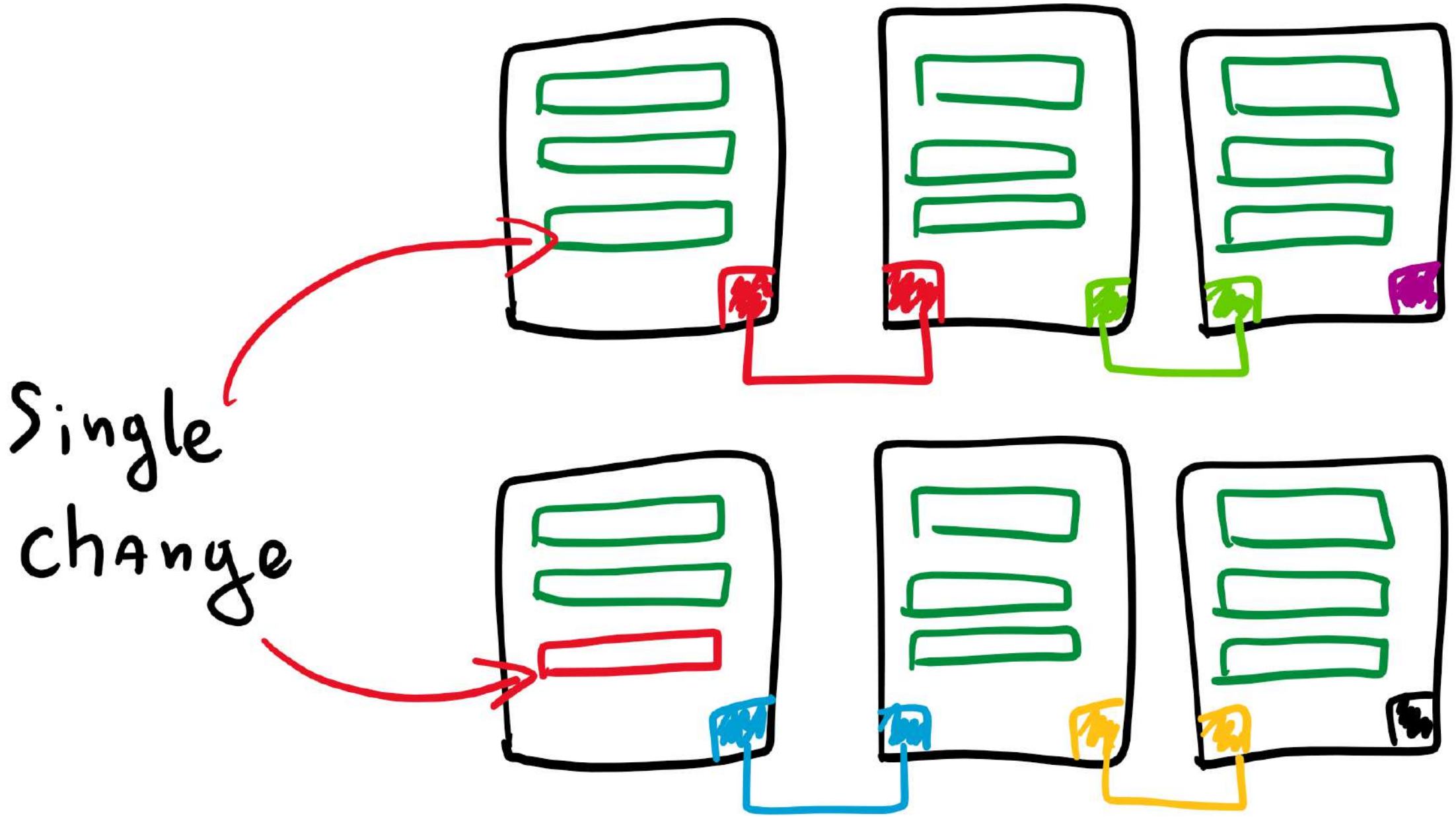
```
// …
// TODO: Implement Signatures
// …
```

Achieving Immutability




Hash




Single
change

```java
public class Block {
	private final List<Transaction> transactions;
	private final String prevBlockHash;
	private final String blockHash;

	// constructor, getters, setters, toString...
}
```

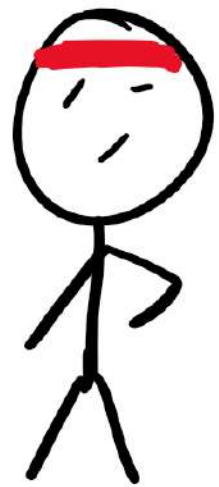
Achieving Consensus

What should we watch?

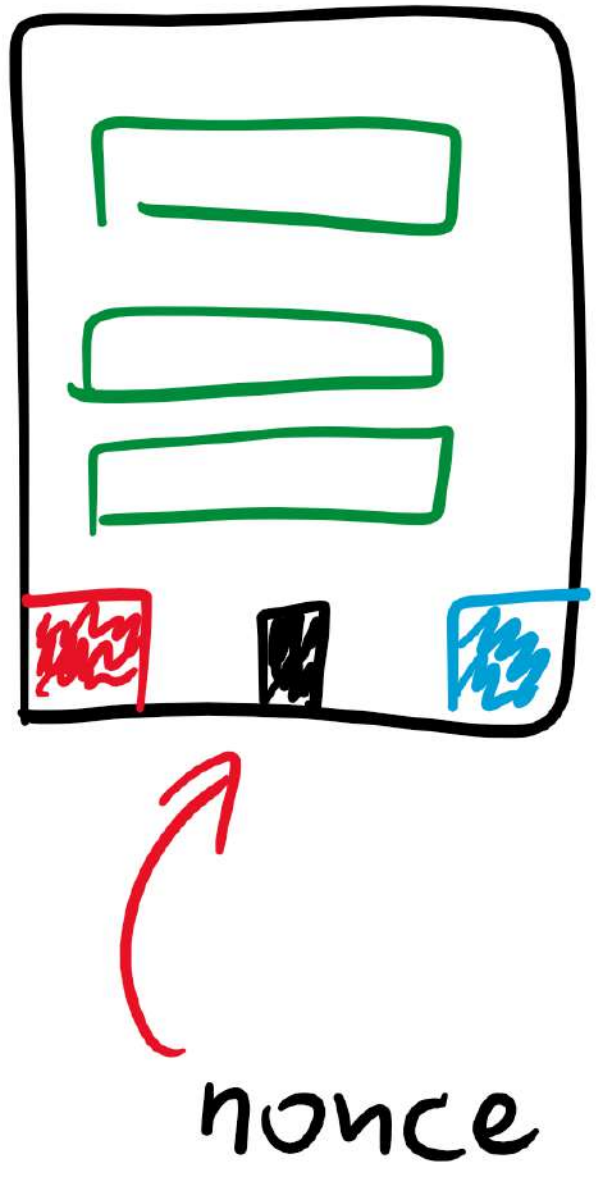
Push-ups Competition!

Score

Yellow	20
Green	23
Red	35




Winner!


nonce

$$SHA256(data + nonce) < target$$

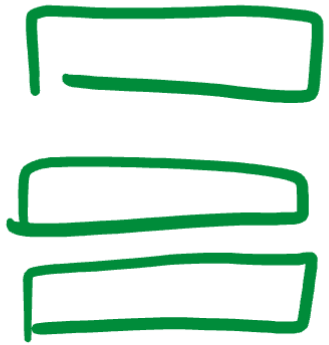
Prev block hash

Pending TXs

nonce

block hash

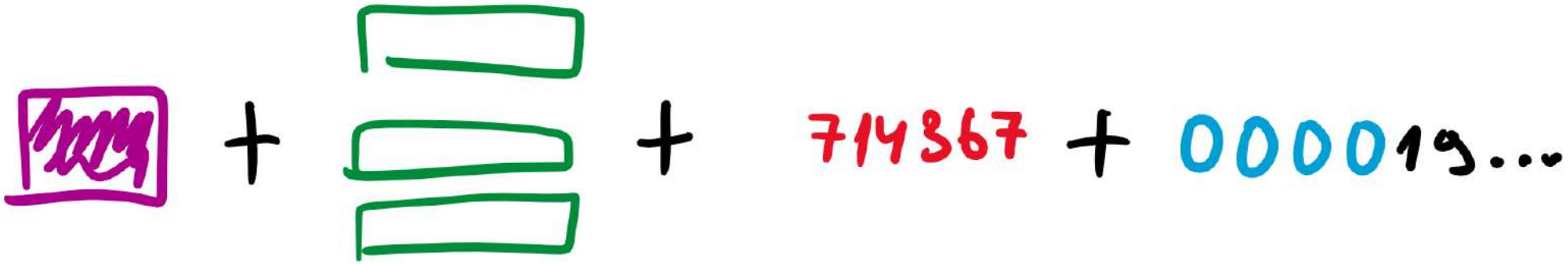
+



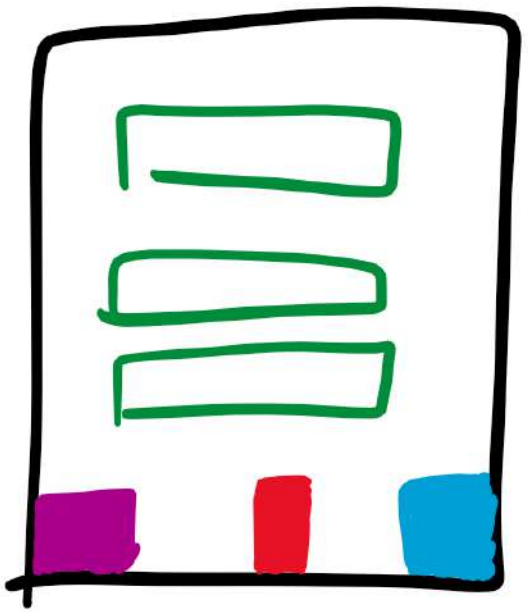
+

nonce	block hash
1	4C1... ✗
2	DAC7... ✗
3	63F... ✗
⋮	⋮
714367	000019... ✓

p-hash TXs nonce block_hash



block



```java
public class Block {
	private final List<Transaction> transactions;
	private final long nonce;
	private final String prevBlockHash;
	private final String blockHash;

	// constructor, getters, setters, toString...
}
```

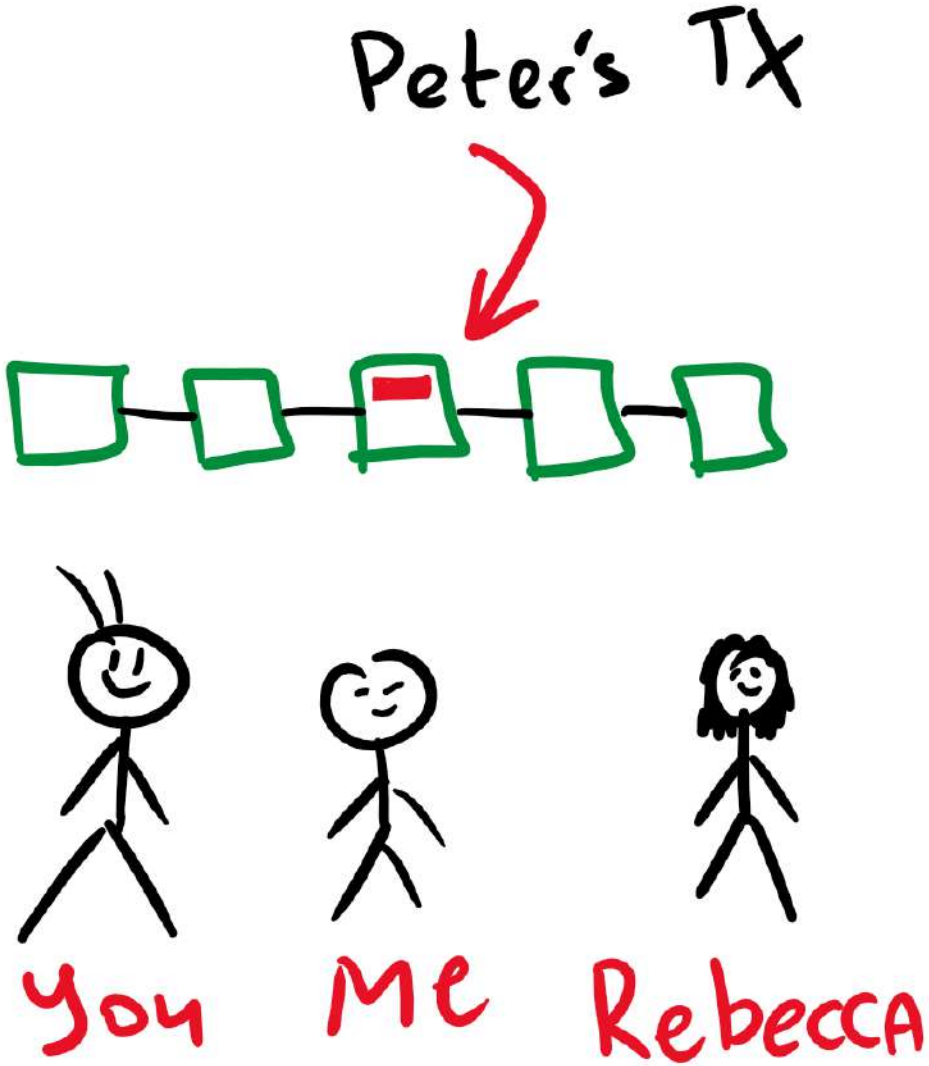
```
public BlockCandidate getMiningJob() {
      String lastBlockHash = blocks.get(blocks.size() - 1)
                  .getBlockHash();

      return new BlockCandidate(
            pendingTransactions, lastBlockHash);
}
```

```
public void submitBlock(Block newBlock) {
      if (isValidBlock(newBlock)) {
            blocks.add(newBlock);
      } else {
            throw new IllegalArgumentException("Invalid block");
      }
}
```

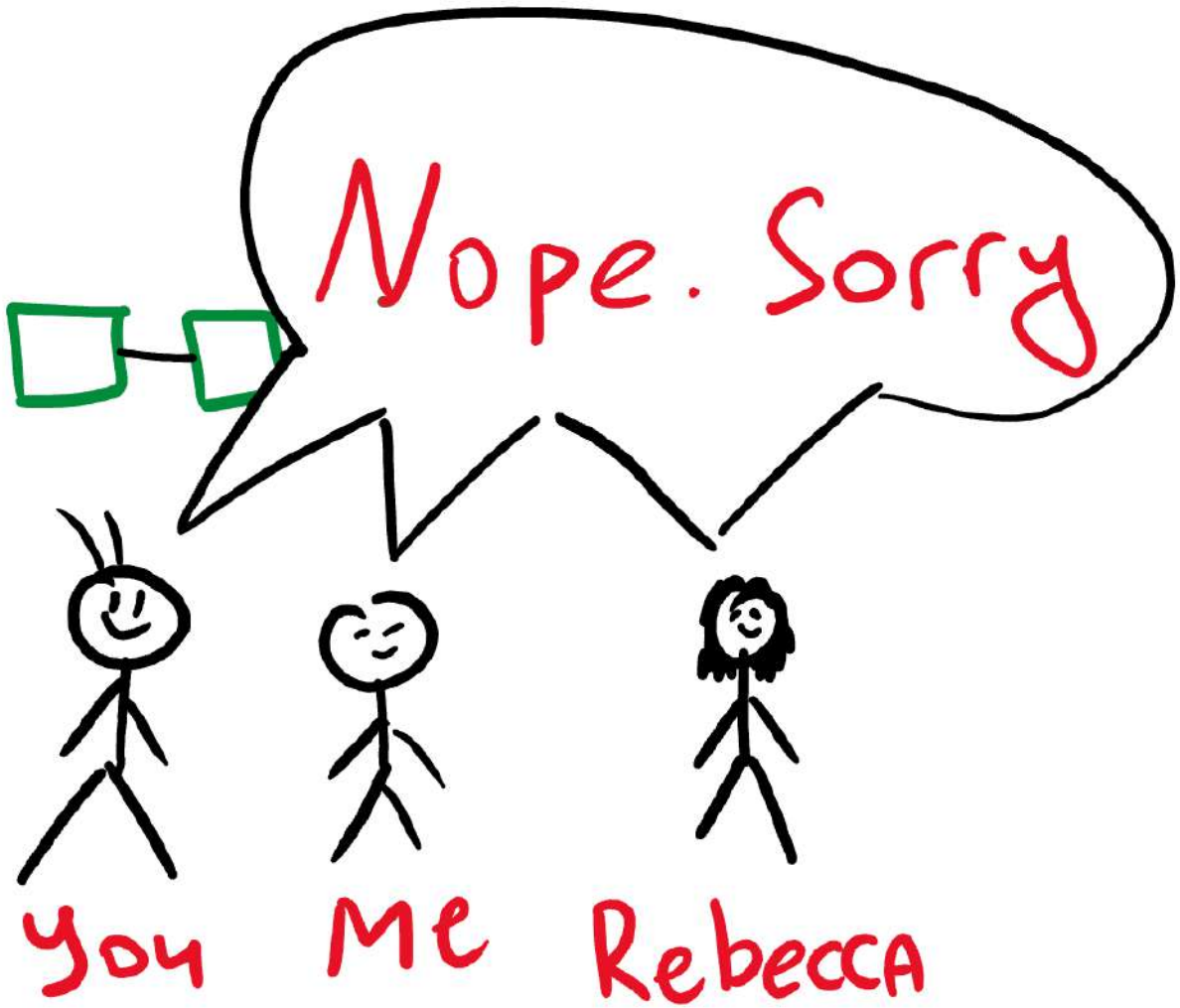
```
private boolean isValidBlock(Block block) {
        // …
        // intermediary calculations…
        // …

        return prevBlockHash.equals(lastBlock.getBlockHash()) &&
                calculatedBlockHash.equals(blockHash) &&
                Utils.startsWithZeroes(blockHash, difficulty);
}
```

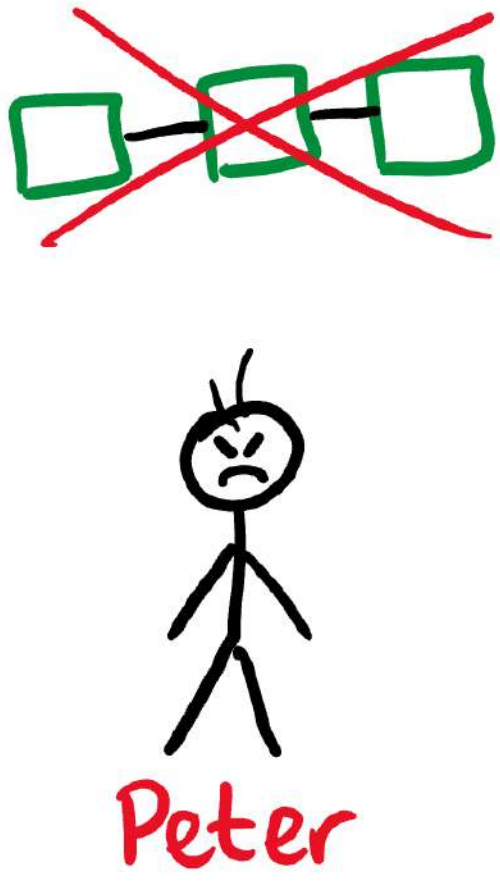
```
public void mine(Blockchain blockchain) {
      // intermediary calculations…

      while (!startsWithZeroes(blockHash, difficulty)) {
            nonce++;
            String blockData =
                  prevBlockHash + transactions + nonce;
            blockHash = Utils.calculateSHA256(blockData);
      }

      // submit mined block…
}
```

The Longest Chain

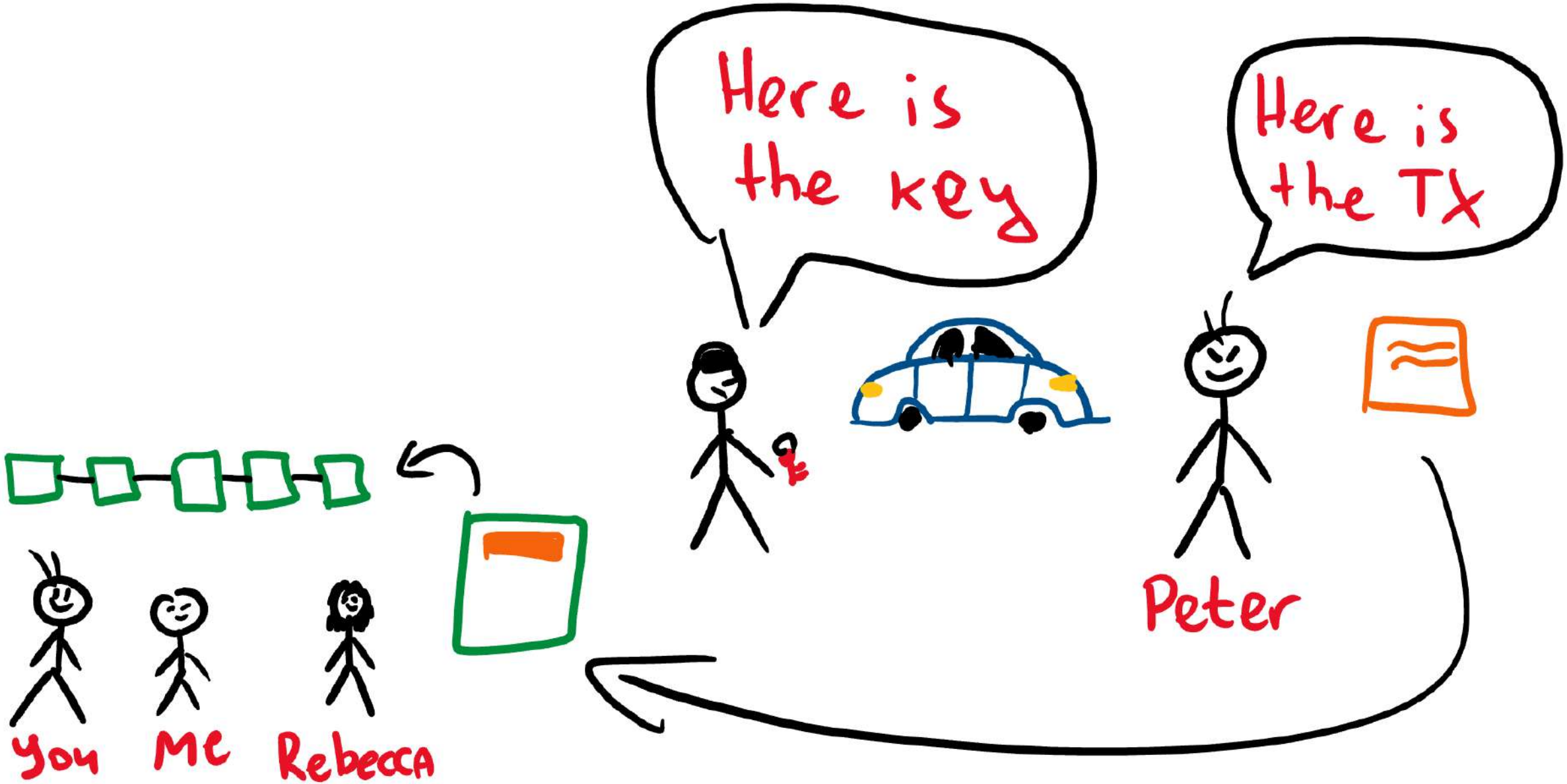

Peter's TX
You
Me
Rebecca

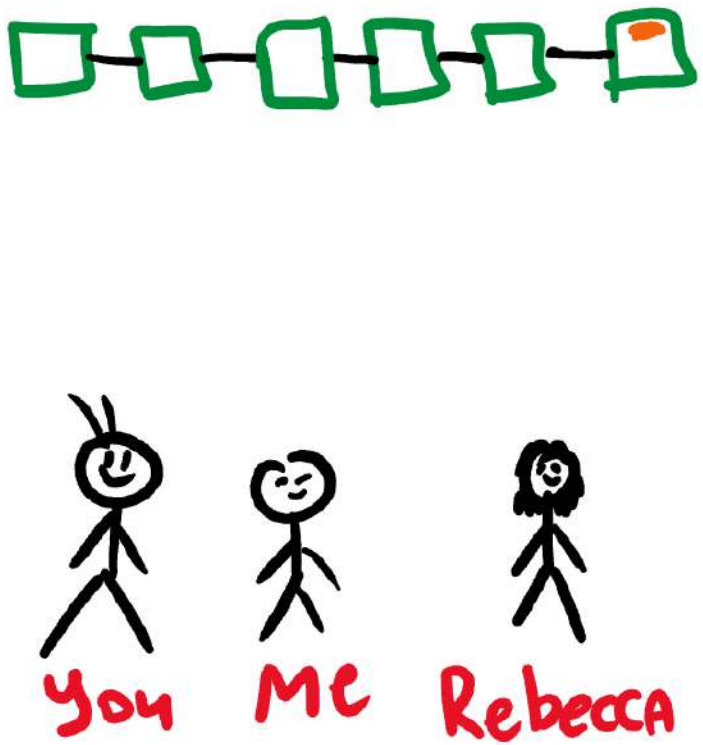

Here is my valid chain
Peter


Nope. Sorry
You
Me
Rebecca

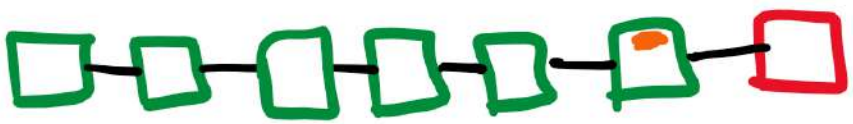

Peter

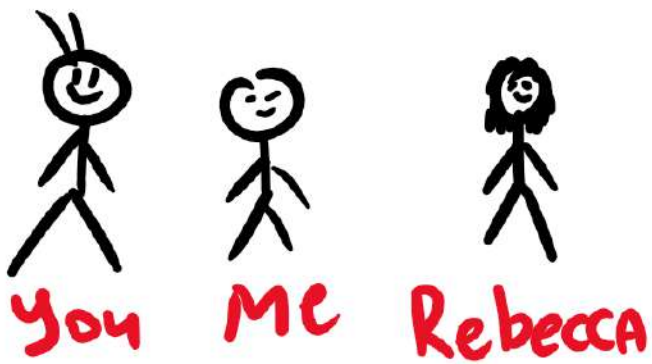
The 51% Attack

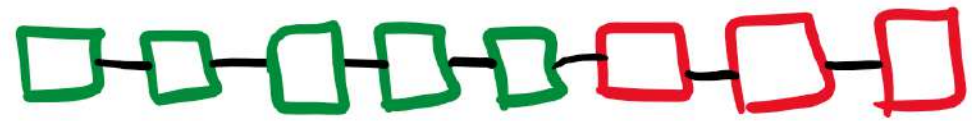

Here is the key
Here is the TX
Peter
You
Me
Rebecca

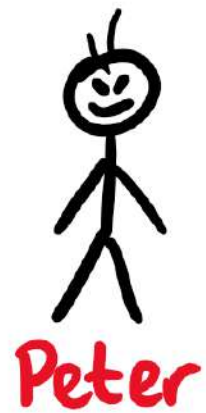

You
Me
Rebecca

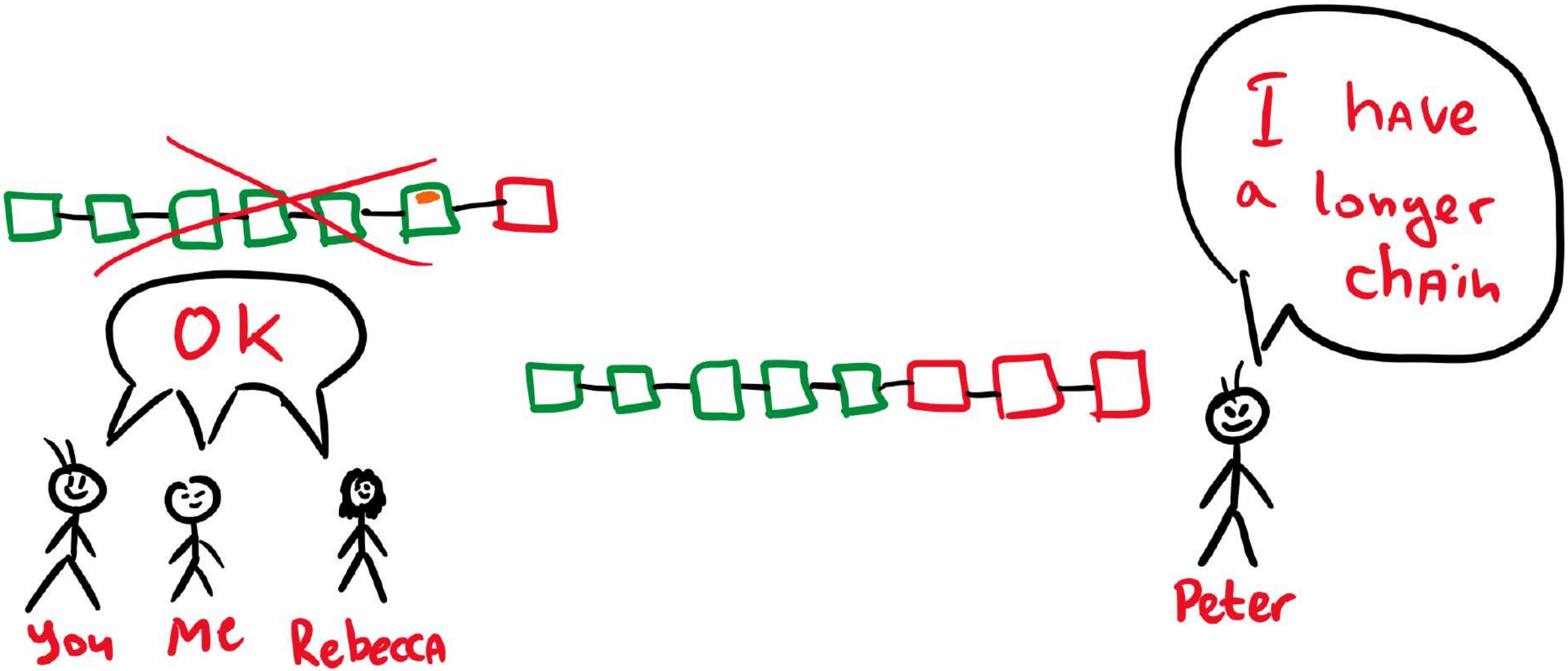

Peter

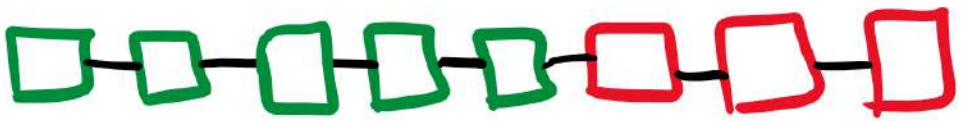


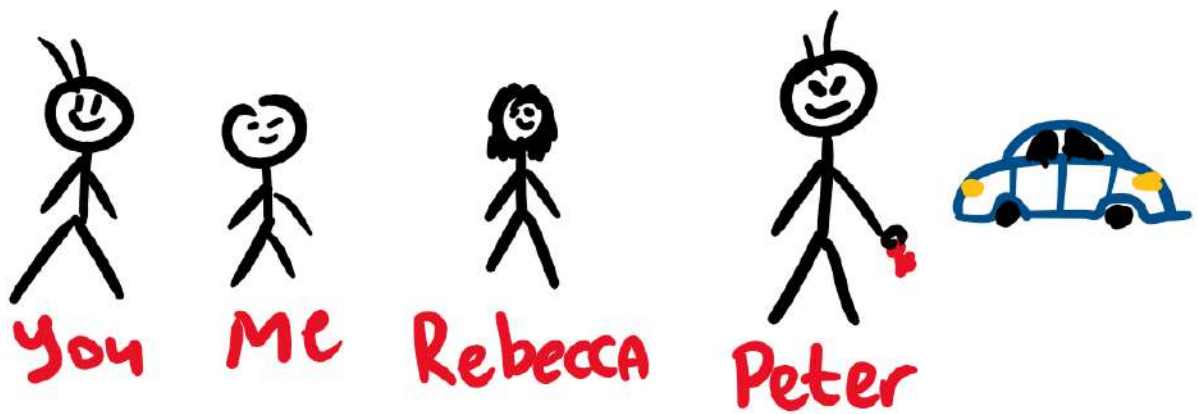

You
Me
Rebecca



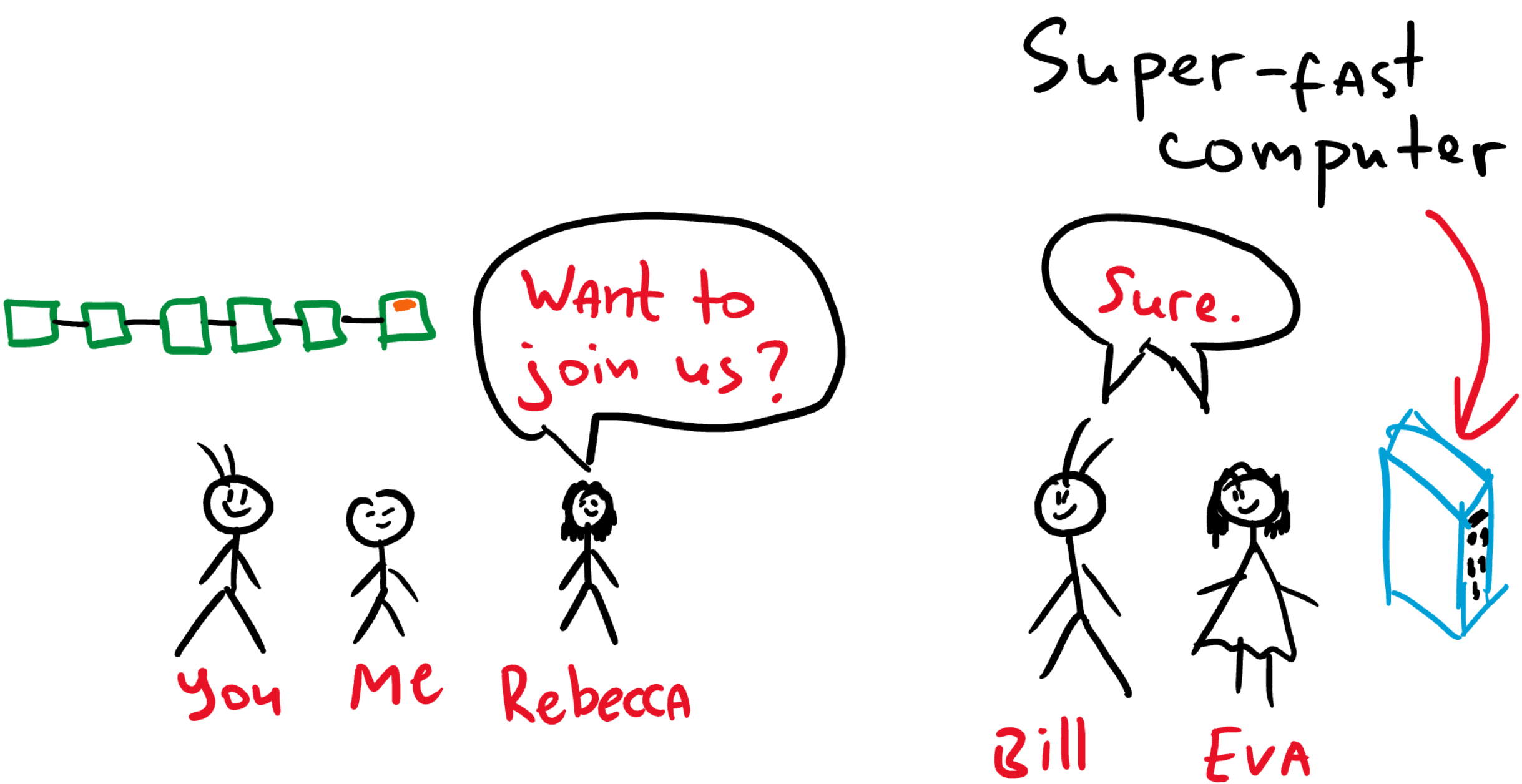

Peter


I have a longer chain
Ok
You
Me
Rebecca
Peter

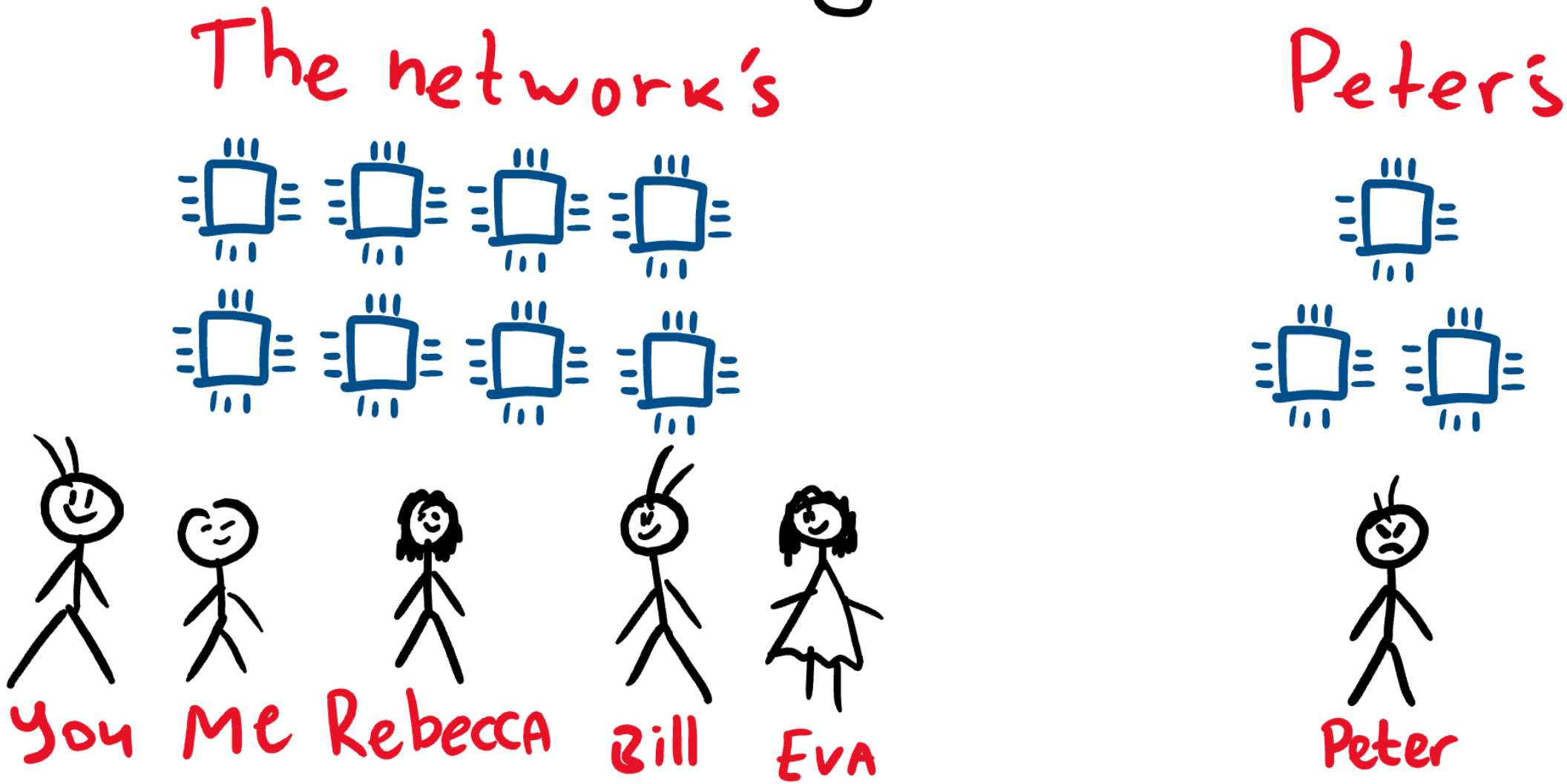



You
Me
Rebecca
Peter

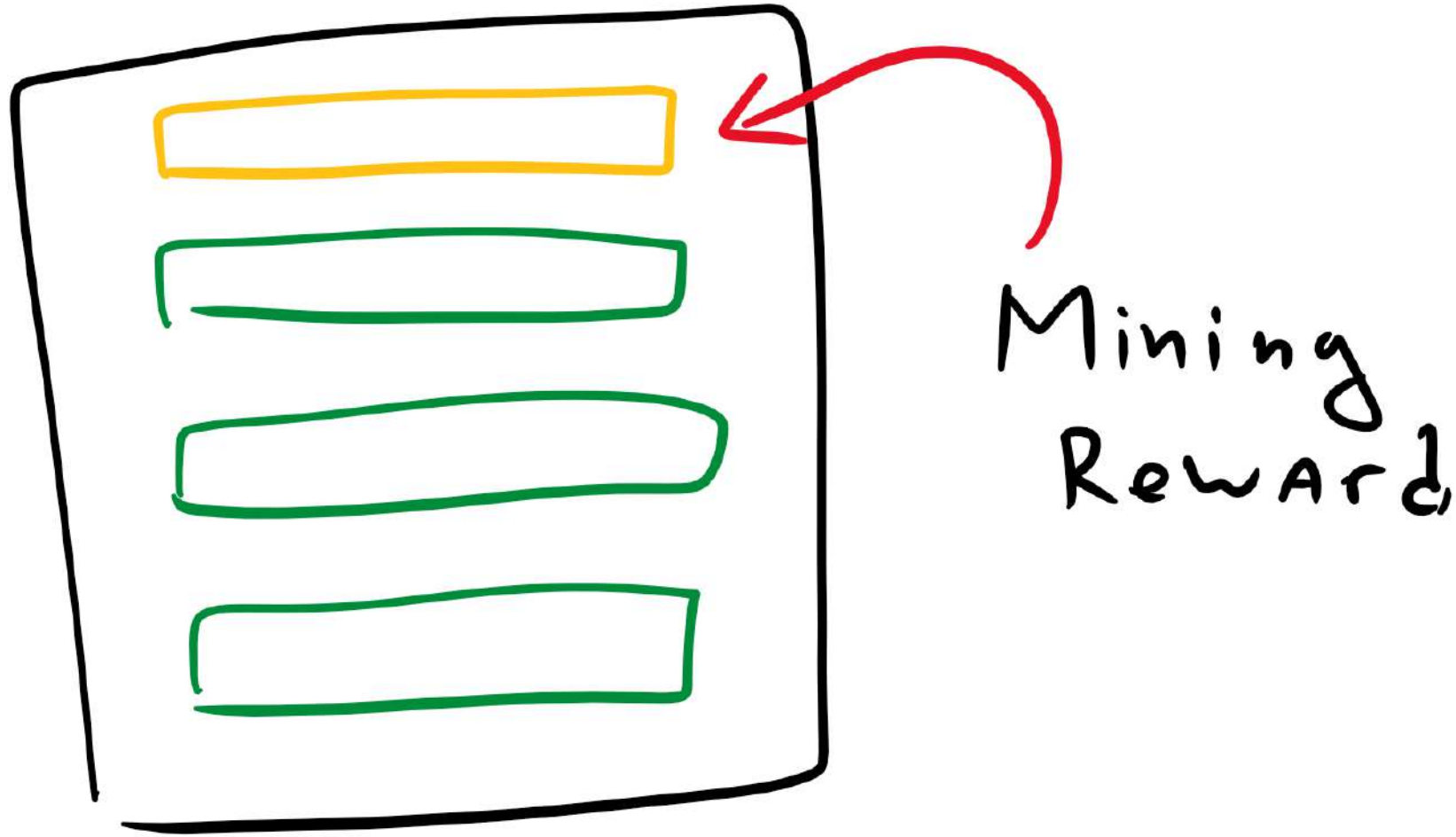

PC


Super-fast computer
Want to join us?
Sure.
You
Me
Rebecca
Bill
Eva

Hashing Power



```java
public class Blockchain {
	private List<Block> blocks;
	private List<Transaction> pendingTransactions;
	private int difficulty = 4;

	// constructors, getters, setters, toString…
}
```


Mining
Reward

```
// …
// TODO: Implement Mining Rewards
// …
```

Demo Time

Get the APP:

/preslavmihaylov/noobchain

Part III:

Beyond finance

Decentralized

Decentralized

Money

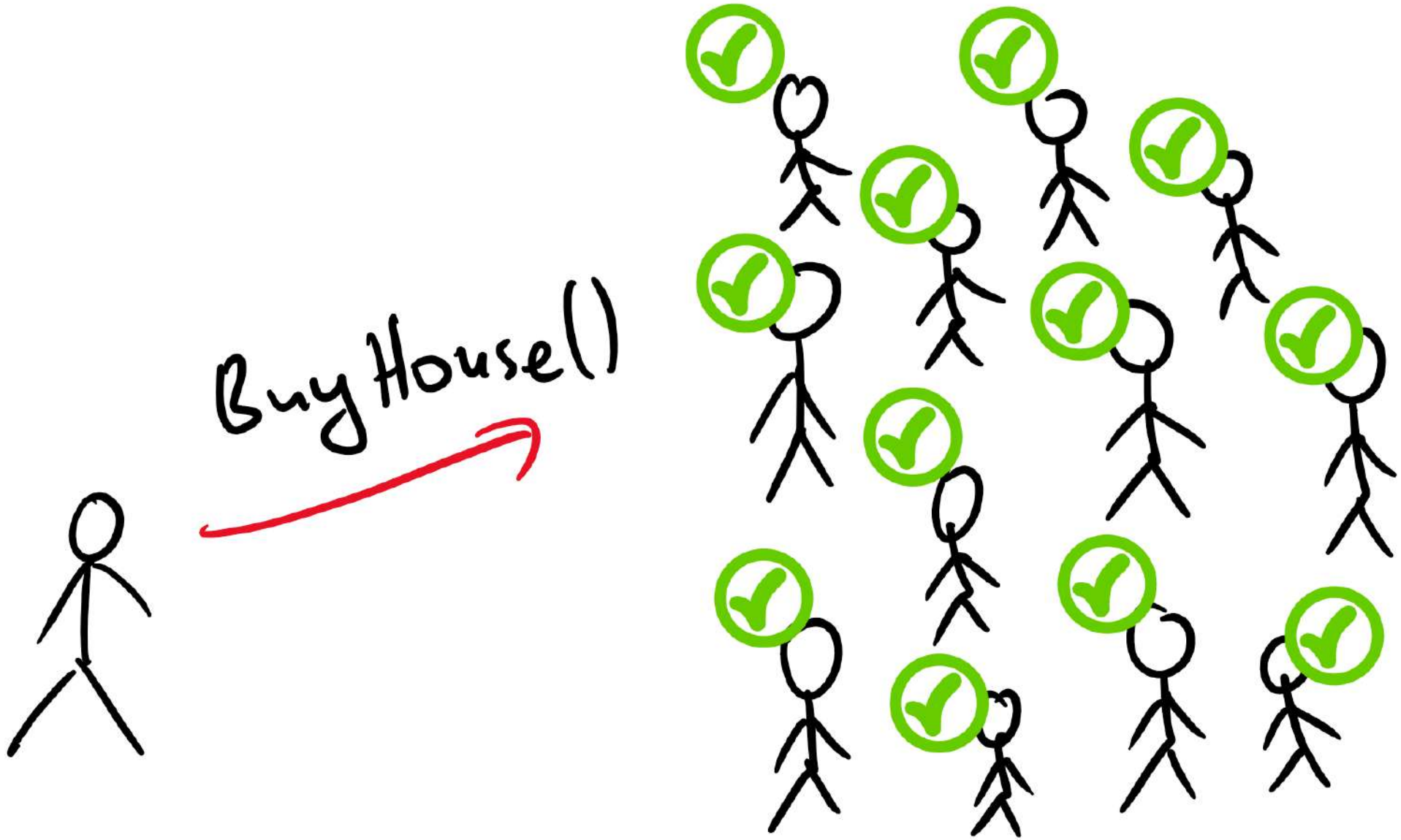
Decentralized

Money → Data

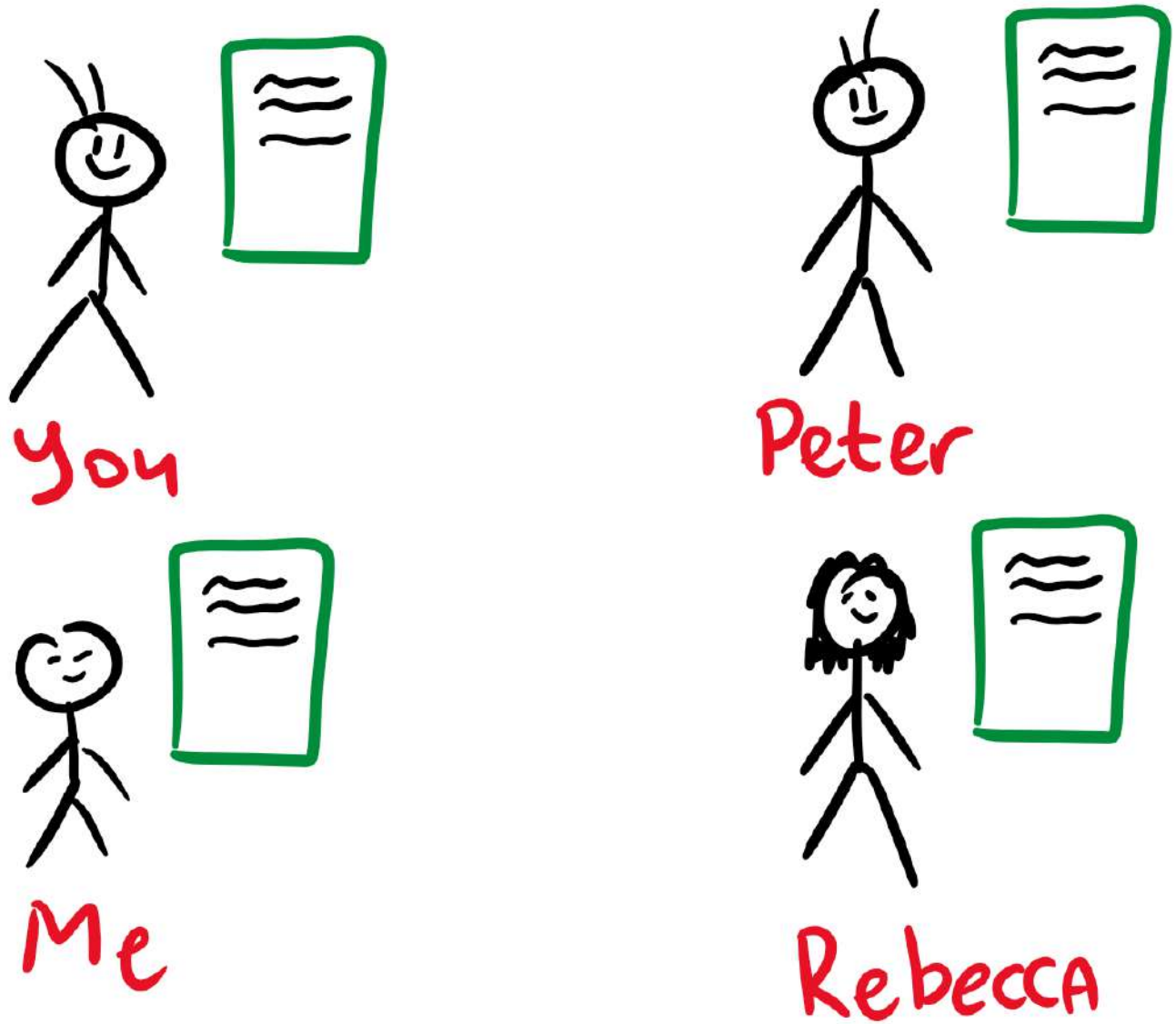
Decentralized

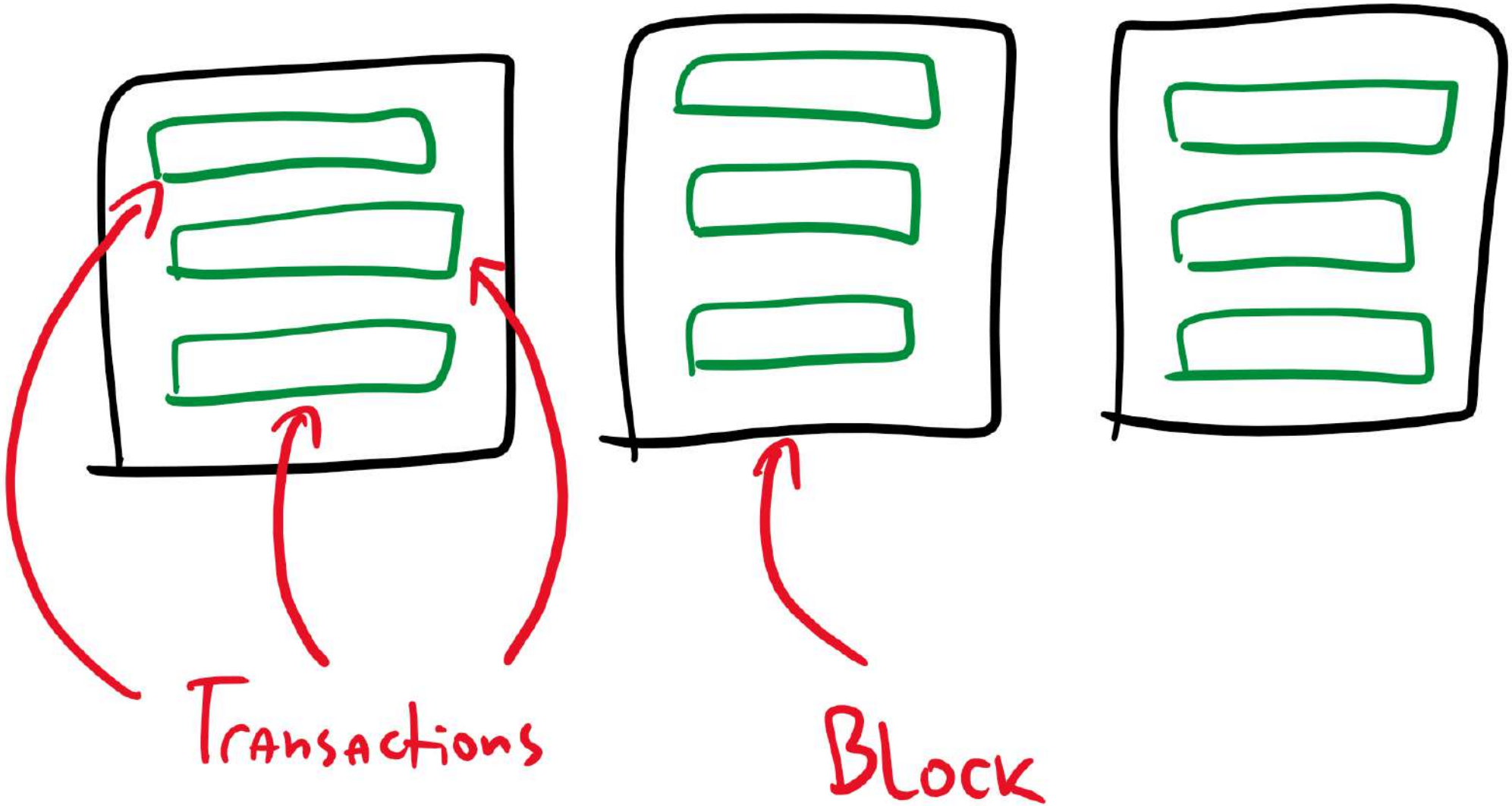
Money → Data → Code

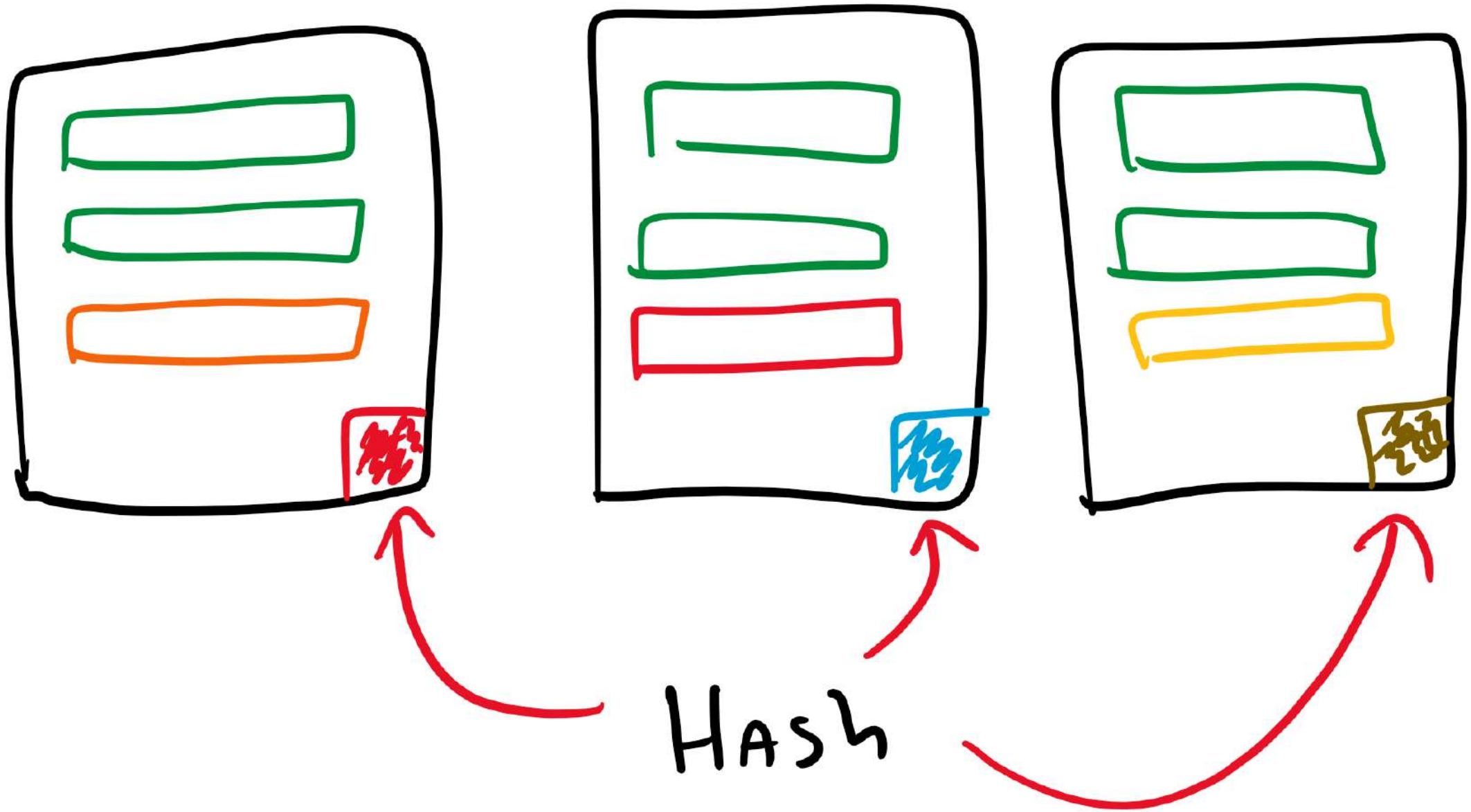
Smart Contract

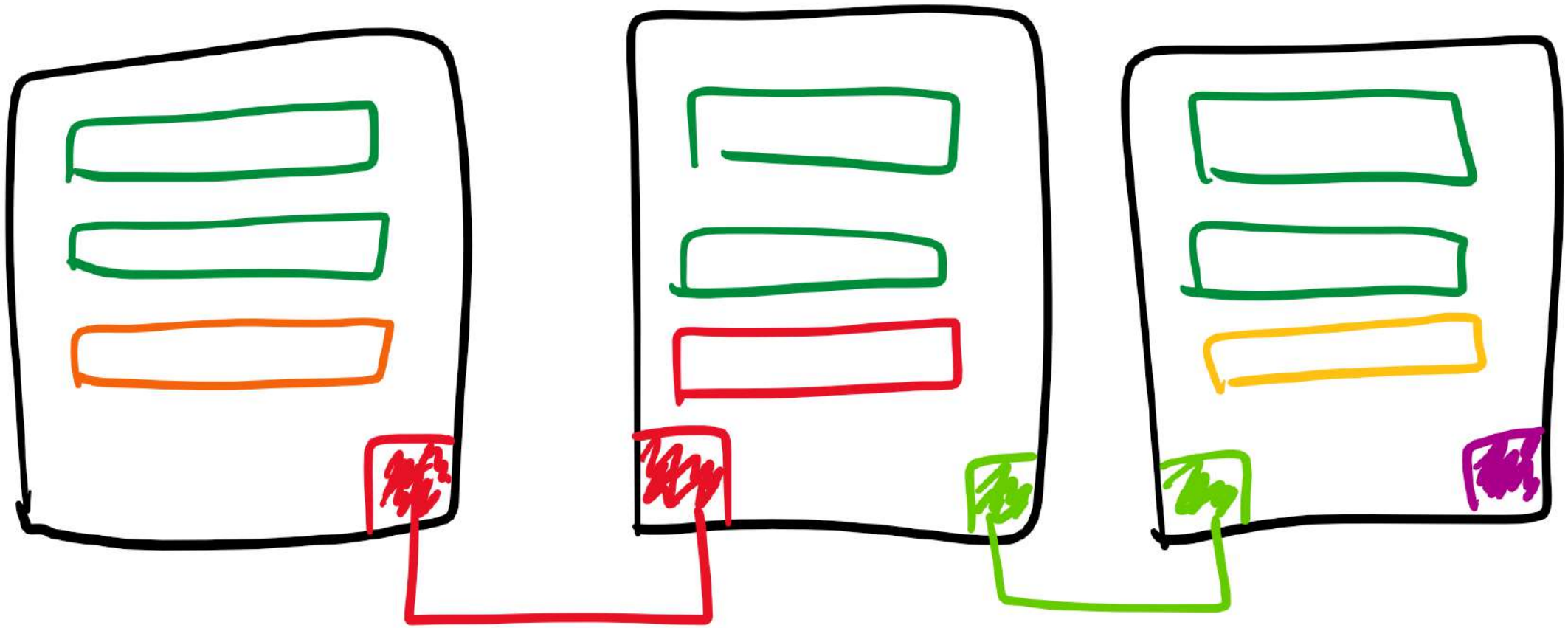

BuyHouse()

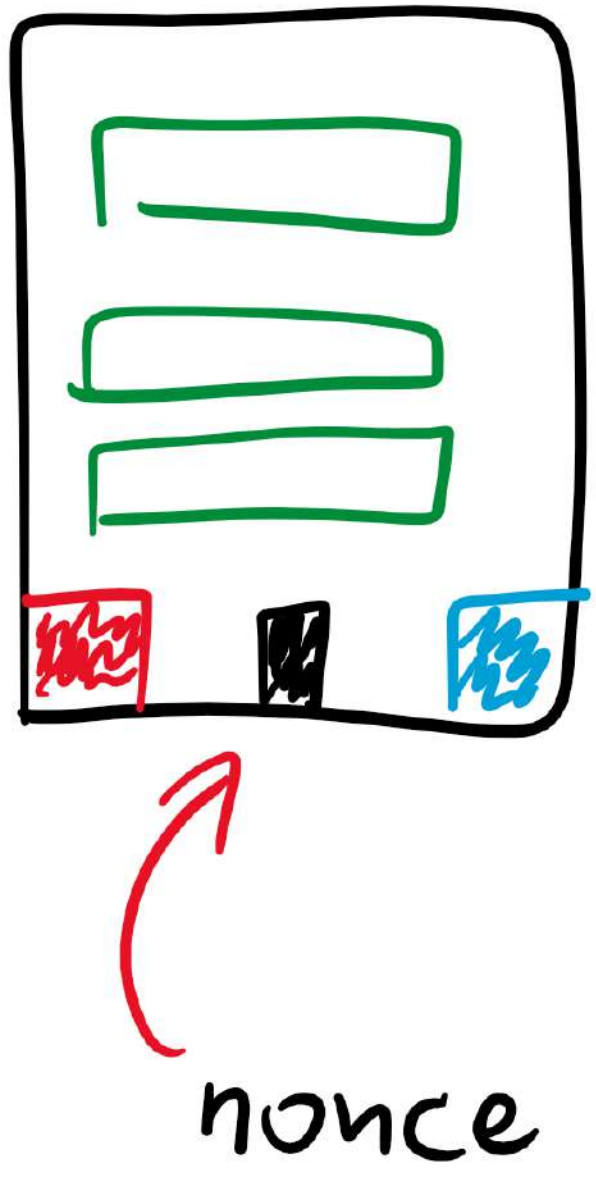
In sum...


You
Peter
Me
Rebecca

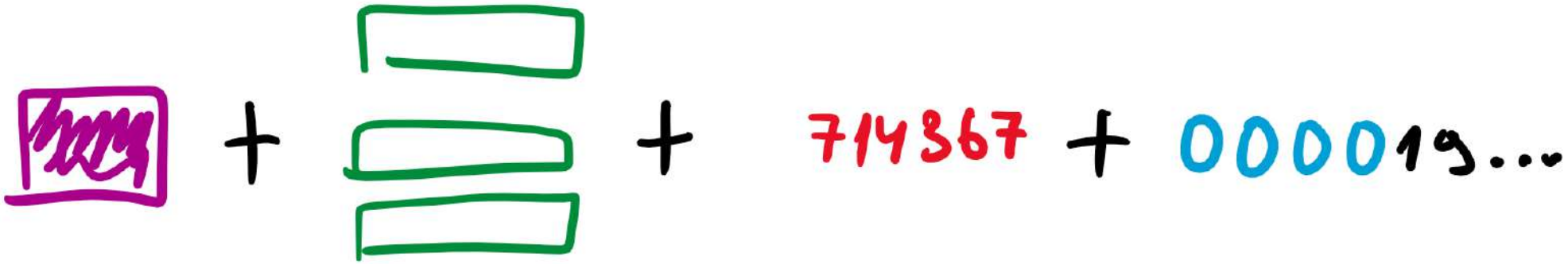

Transactions
Block


Hash

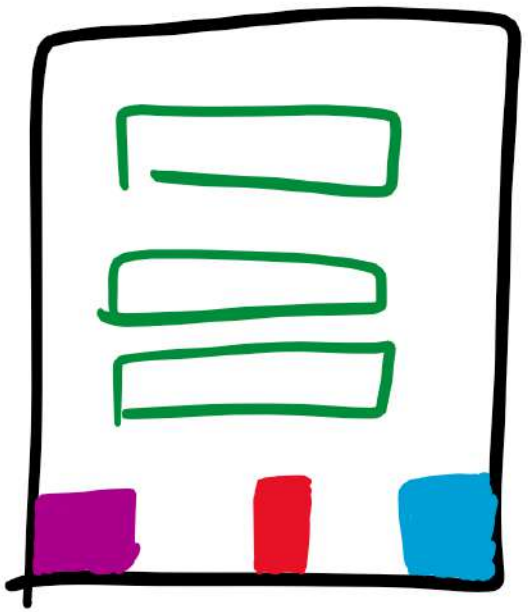



nonce

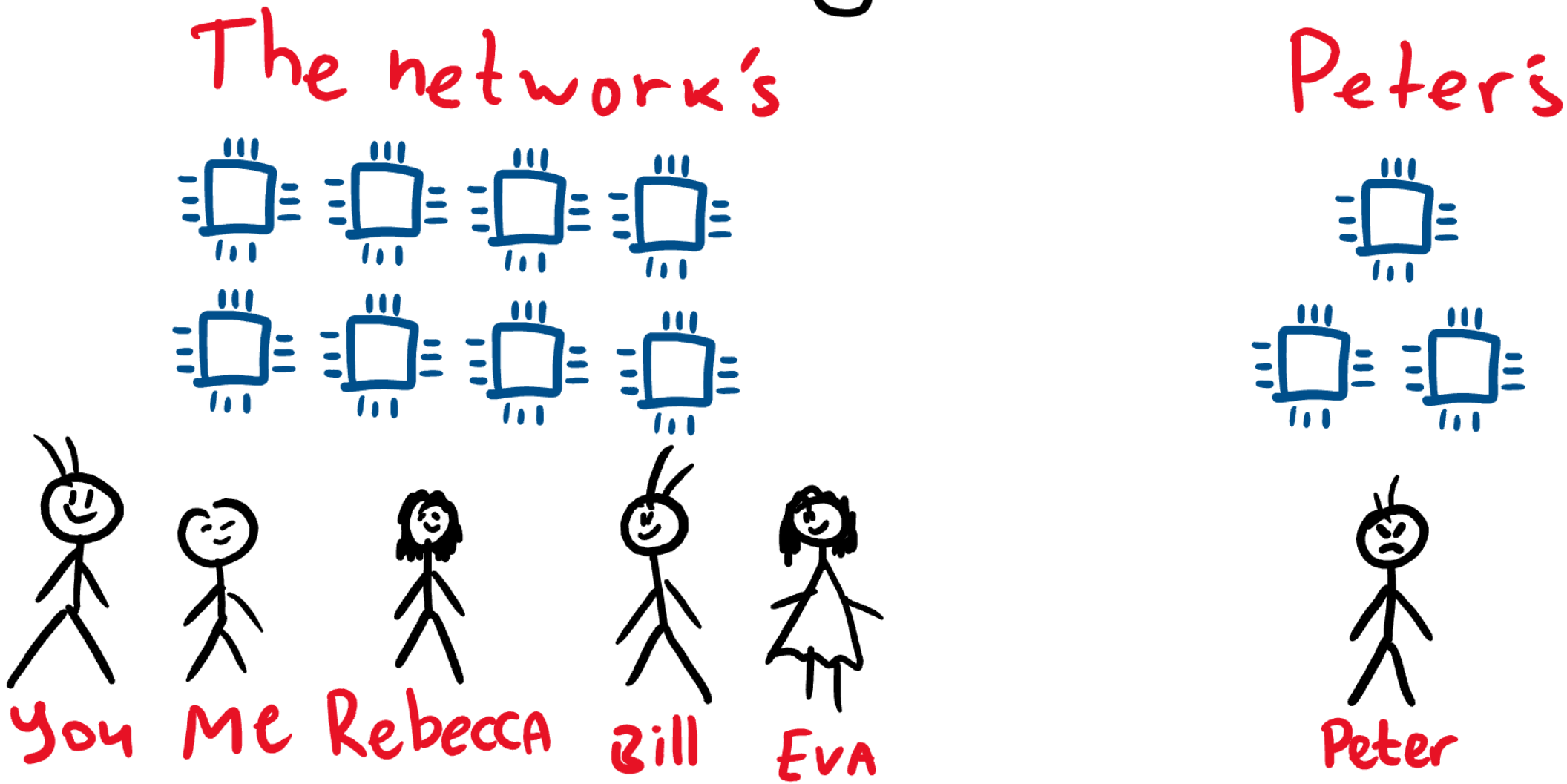
p-hash TXs nonce block_hash



block



Hashing Power

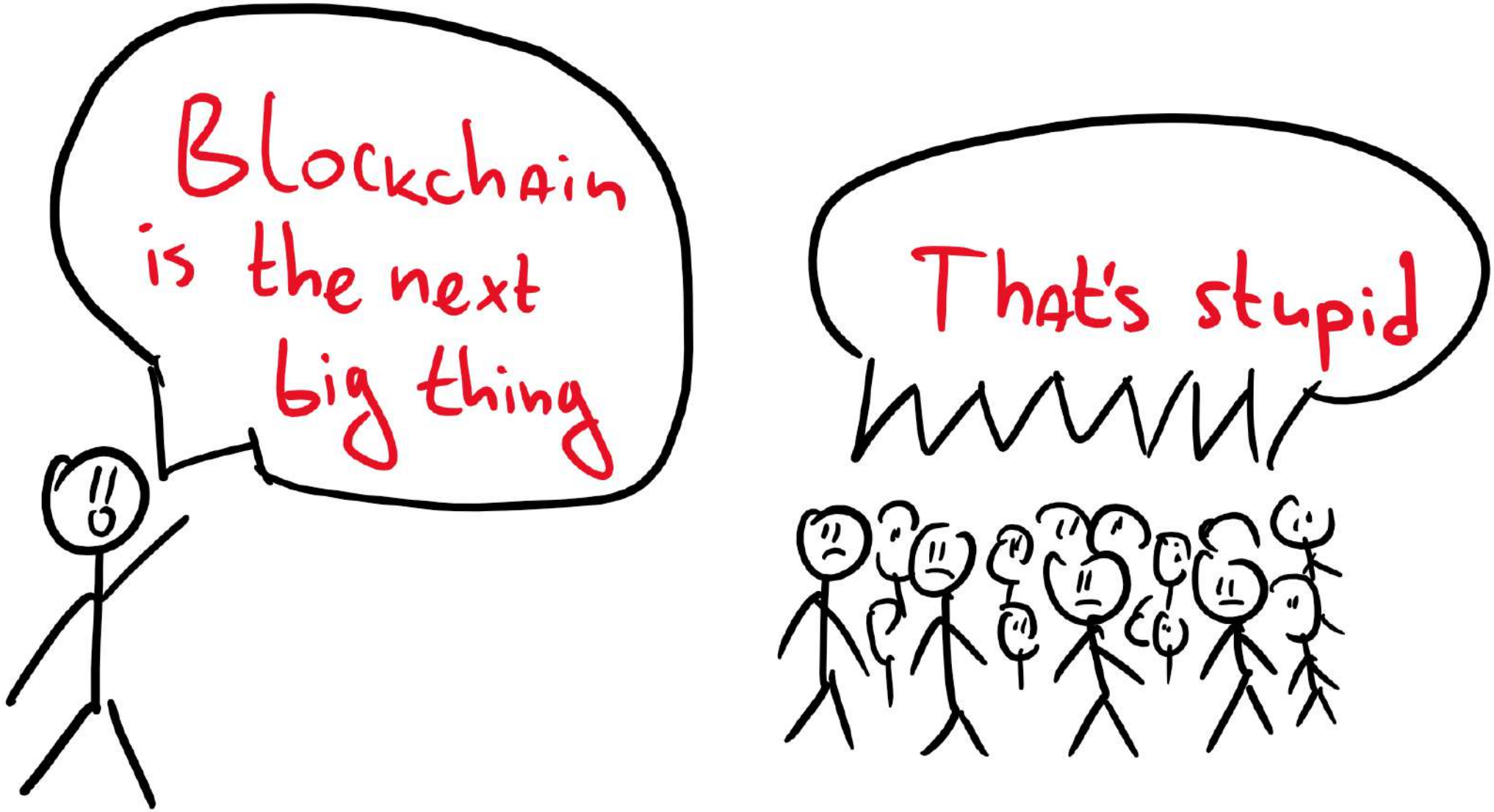


Decentralized

Money → Data → Code

It all sounds great but...

Blockchain is still a


Blockchain is the next big thing
That's stupid


The earth is round
That's stupid

Now is the time for

Pioneers

Now is the time for

Pioneers

Preslav Mihaylov

/preslavmihaylov