Blockchain Technology and Applications

CS 989

Smart contracts and p2p network

Dr. Ir. Angshuman Karmakar

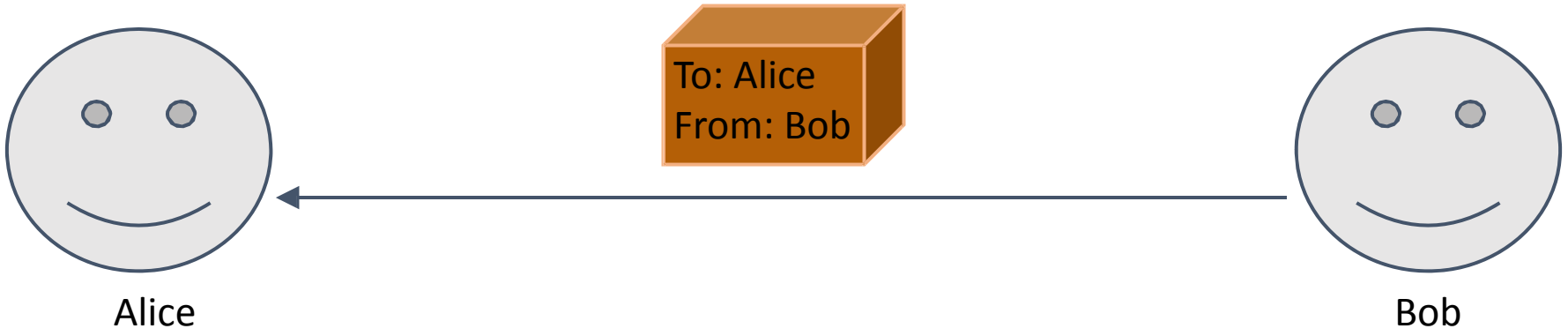
IIT Kanpur

Teaching assistants

- **Sumit Lahiri** (sumitl@cse.iitk.ac.in)
- **Chavan Sujeet** (sujeetc@cse.iitk.ac.in)

Smart Contracts

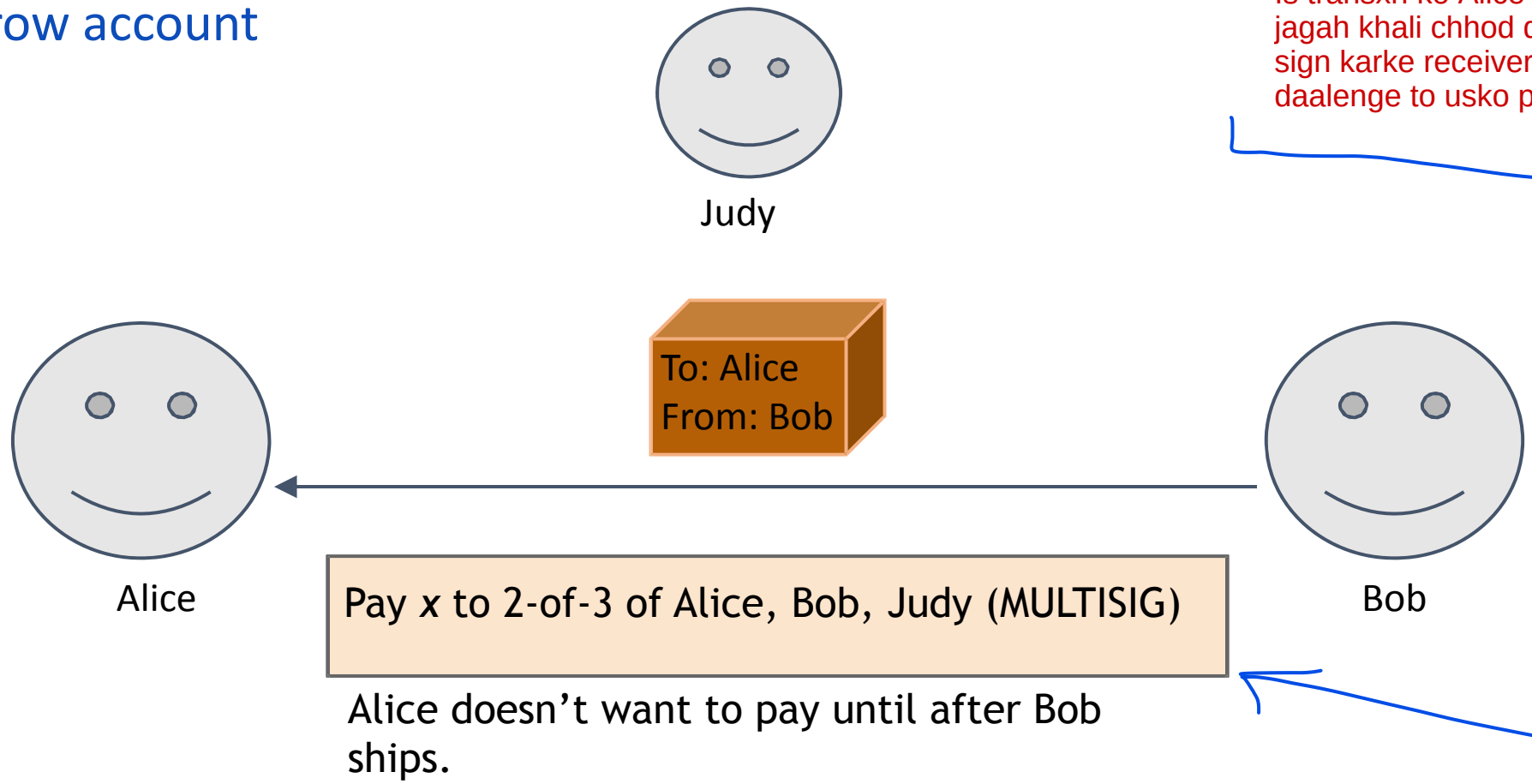
Escrow account



- Alice doesn't want to pay until she has received the good
- Bob doesn't want to send the good until Alice pays

Smart Contracts

Escrow account

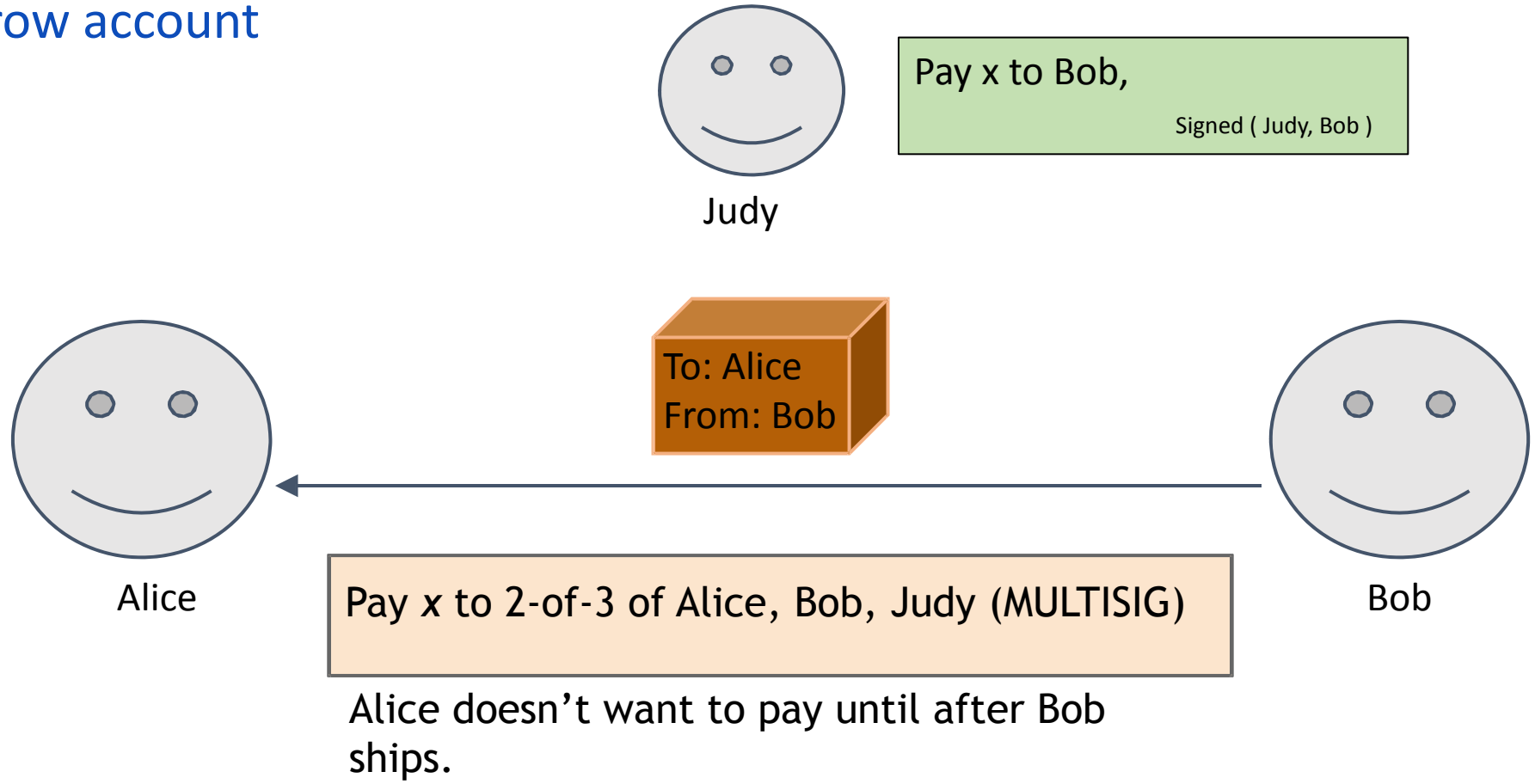


Is transxn ko Alice ne jab bnaya tab receiver ki jagah khali chhod di, and ab jab koi bhi do log sign karke receiver ki jagah kisiko bhi daalenge to usko payment hojayegi

- Involves a third-party trusted by both
- Create a 2-to-3 multi-signature

Smart Contracts

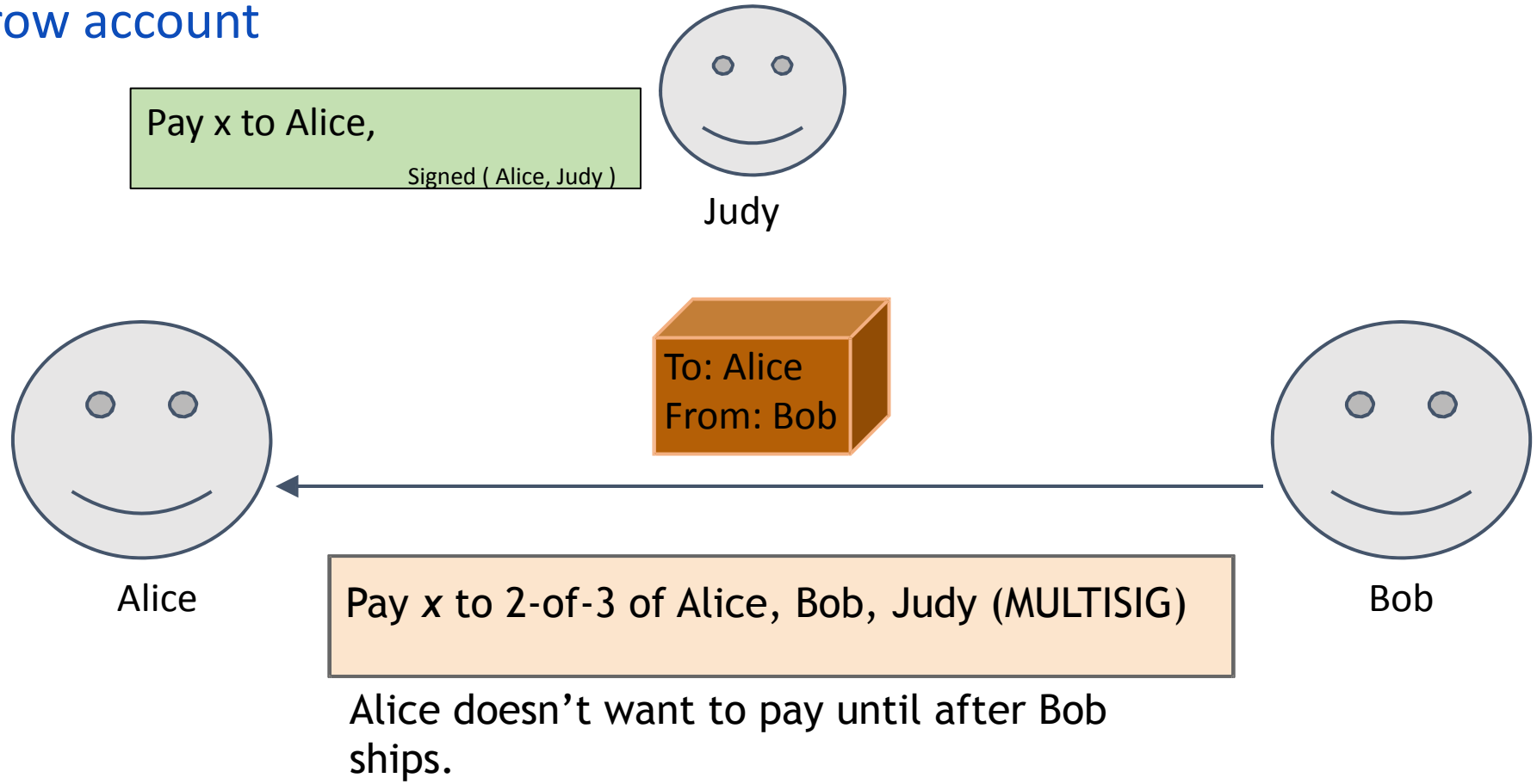
Escrow account



- In the normal case, Alice receives the case
  - Bob gets paid

Smart Contracts

Escrow account



Alice doesn't want to pay until after Bob ships.

- In the disputed case, Alice receives the case
- Trustless escrow?

Transactions in Bitcoin

Green account

Alice

Bob

- Alice wants to buy from Bob
- Bob might be offline--> Can't check the transaction
- Bob can't wait for 6 transactions
  - E.g. : Coffee shop, street vendor
- How to stop Alice from double-spending?

Transactions in Bitcoin

Green account

It's me, Alice! Could you make out a green payment to Bob?

Reputed Institution

Mutual understanding

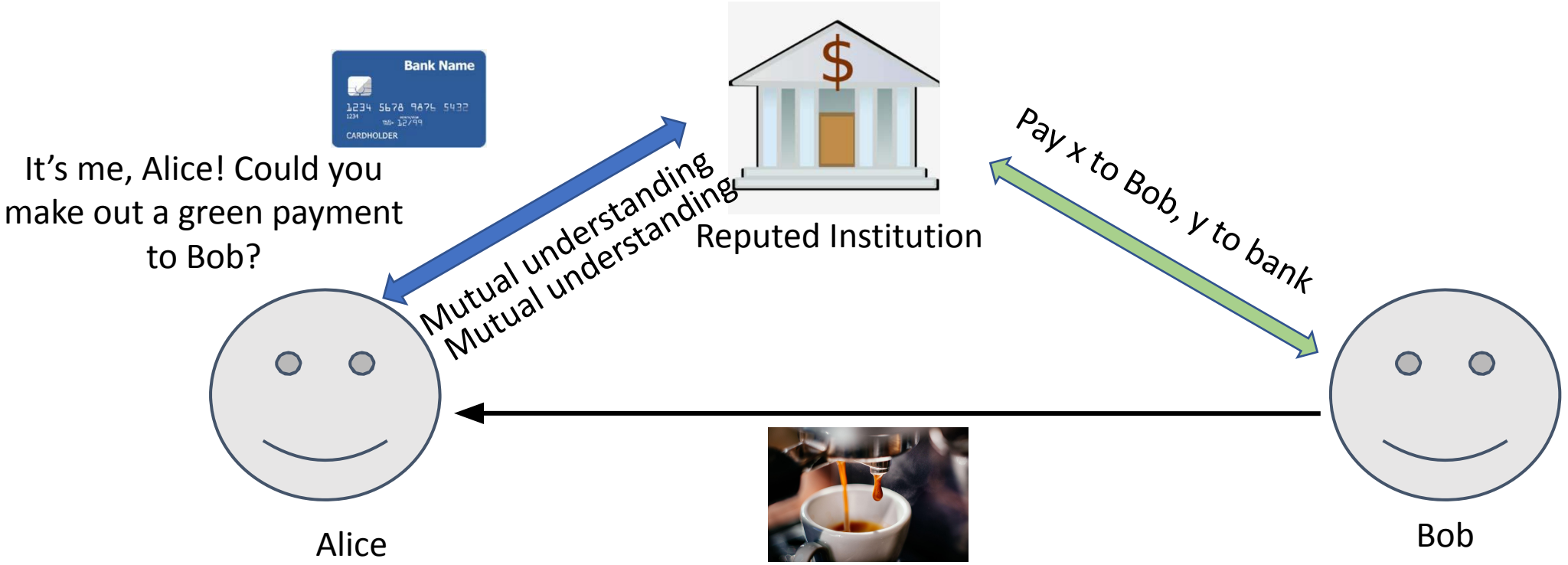
Alice

Bob

- Reputed third-party
- Never double-spends

Transactions in Bitcoin

Green account



- Bank pays Bob from its own account
- Green account
- Bob hands over the merchandise

Transactions in Bitcoin

Micro payments

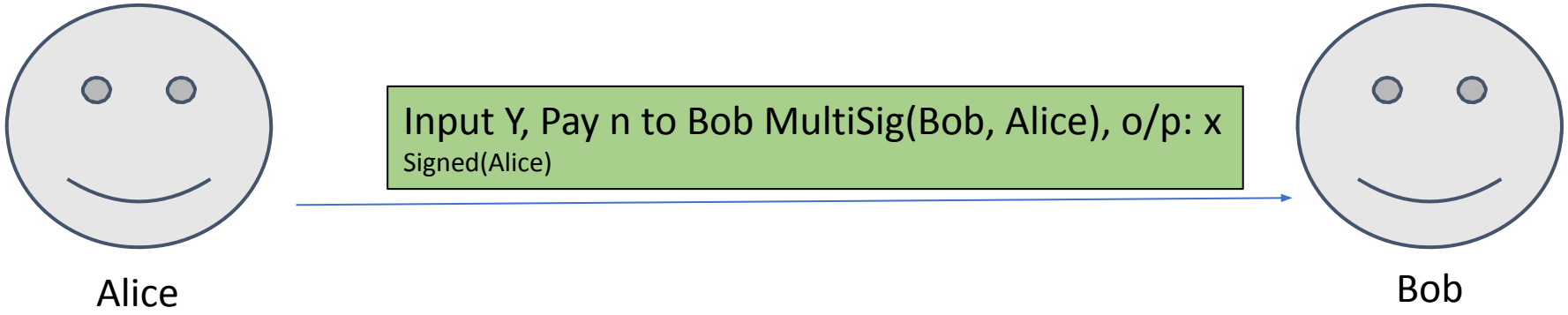
Alice

Bob

- Bob requires Alice to pay continually
  - Small amounts
  - Regular interval
  - E.g. parking service, utility provider, wireless service provider
- Normal transactions?

Transactions in Bitcoin

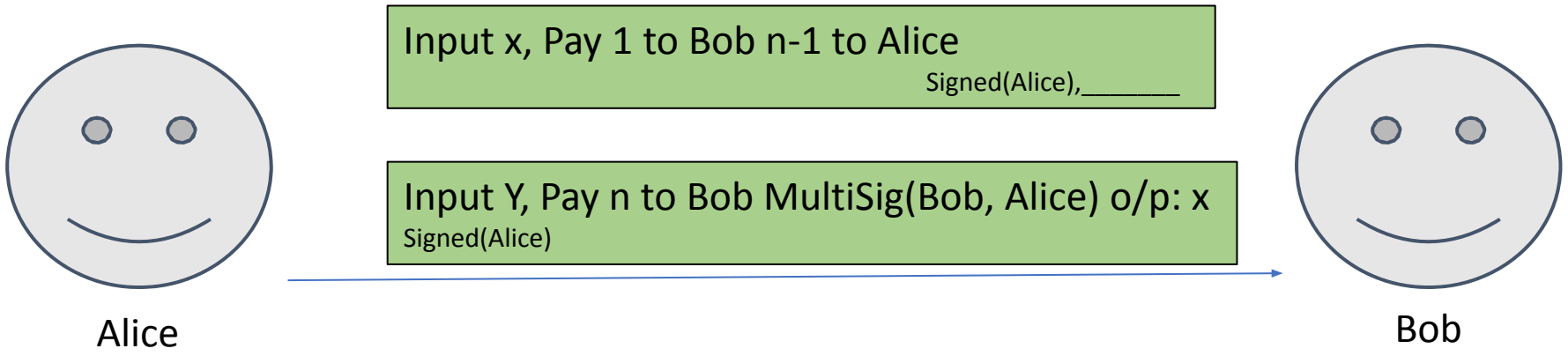
Micro payments



- Alice creates a transaction to Bob
  - An amount > maximum possible use
  - Escrow transaction

Transactions in Bitcoin

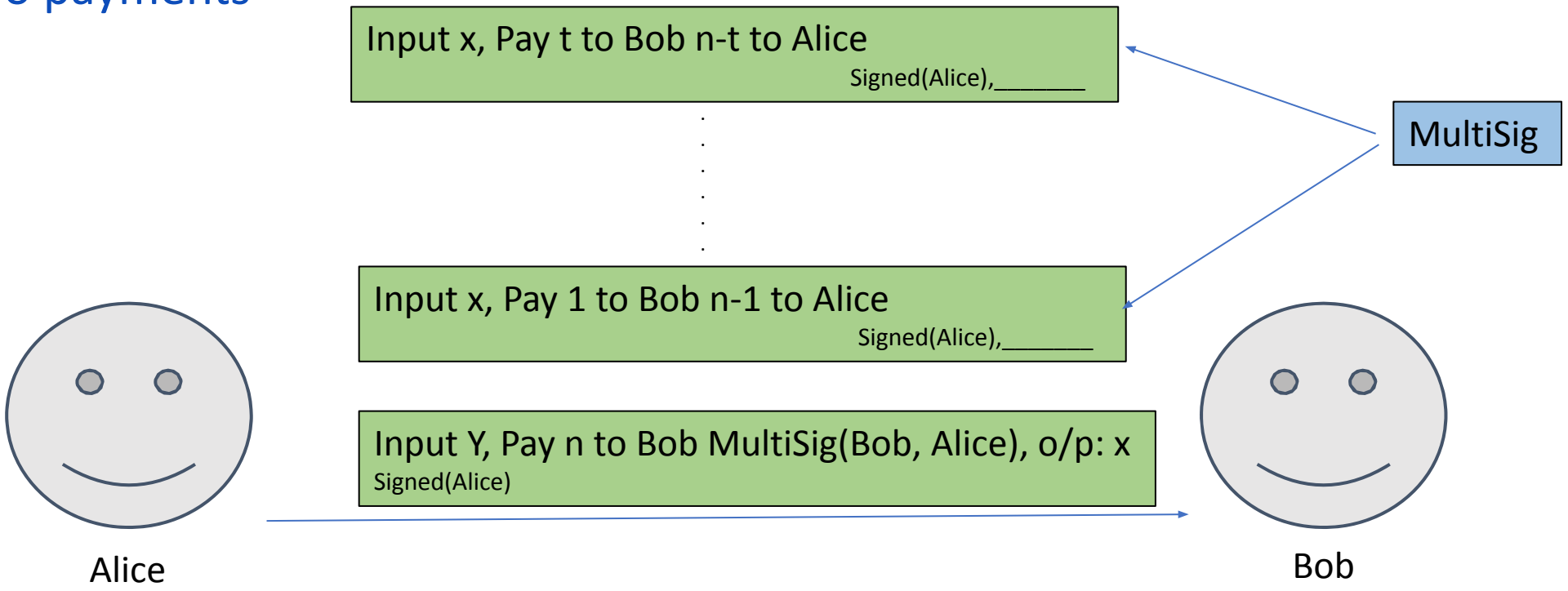
Micro payments



- For each periodical charge
  - Alice creates MULTISIG transactions that pays unit charge to Bob
  - And returns the rest back to her

Transactions in Bitcoin

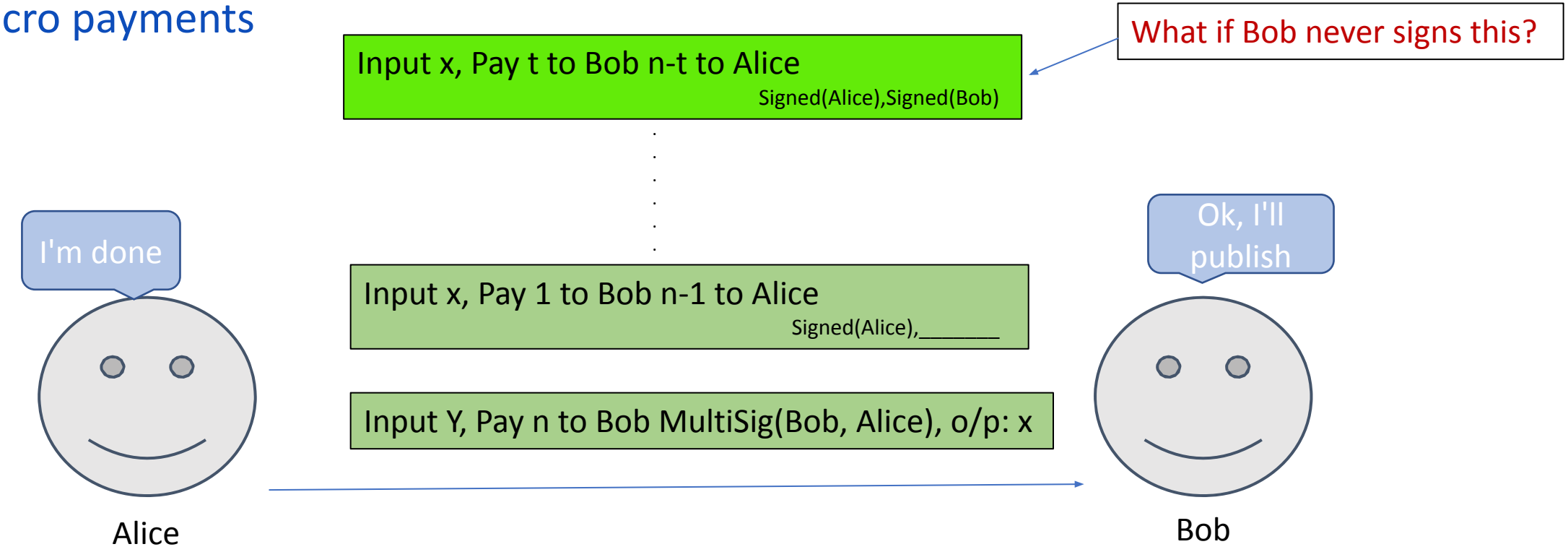
Micro payments



- This goes on
- These communications are on private channel between Bob and Alice
- Bob stores them and doesn't publish them

Transactions in Bitcoin

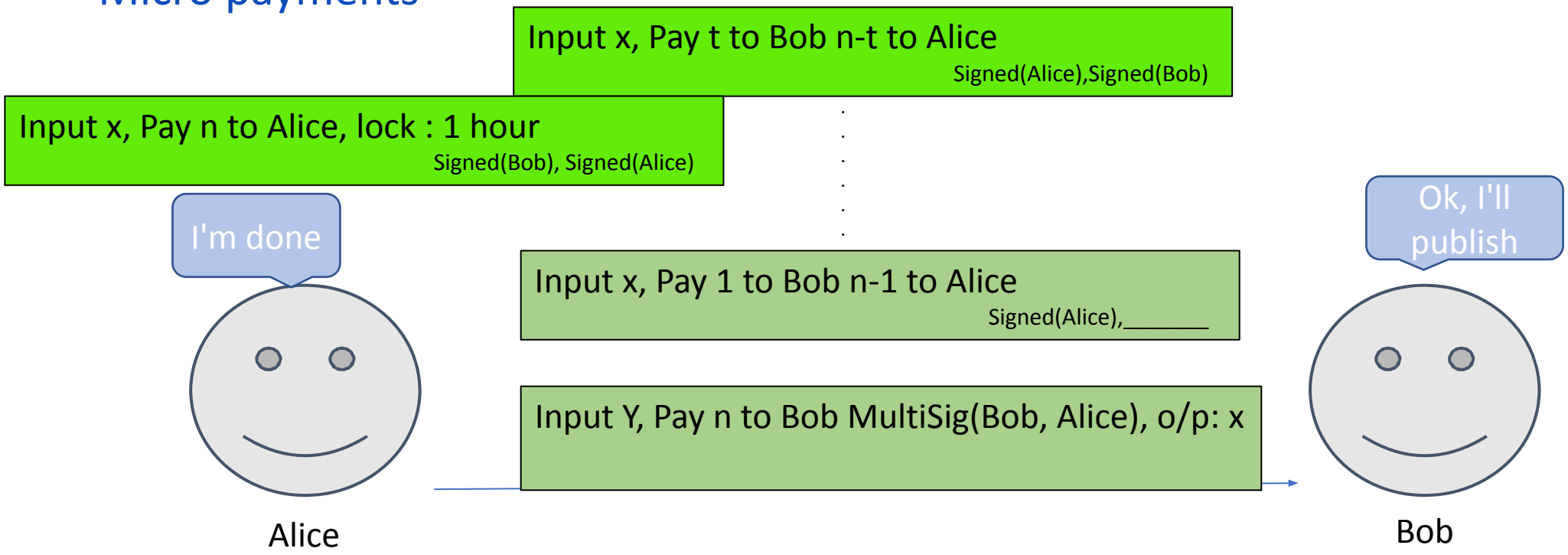
Micro payments



- After Alice is finished, she informs Bob
- Bob signs the final transaction and publishes on the network

Transactions in Bitcoin

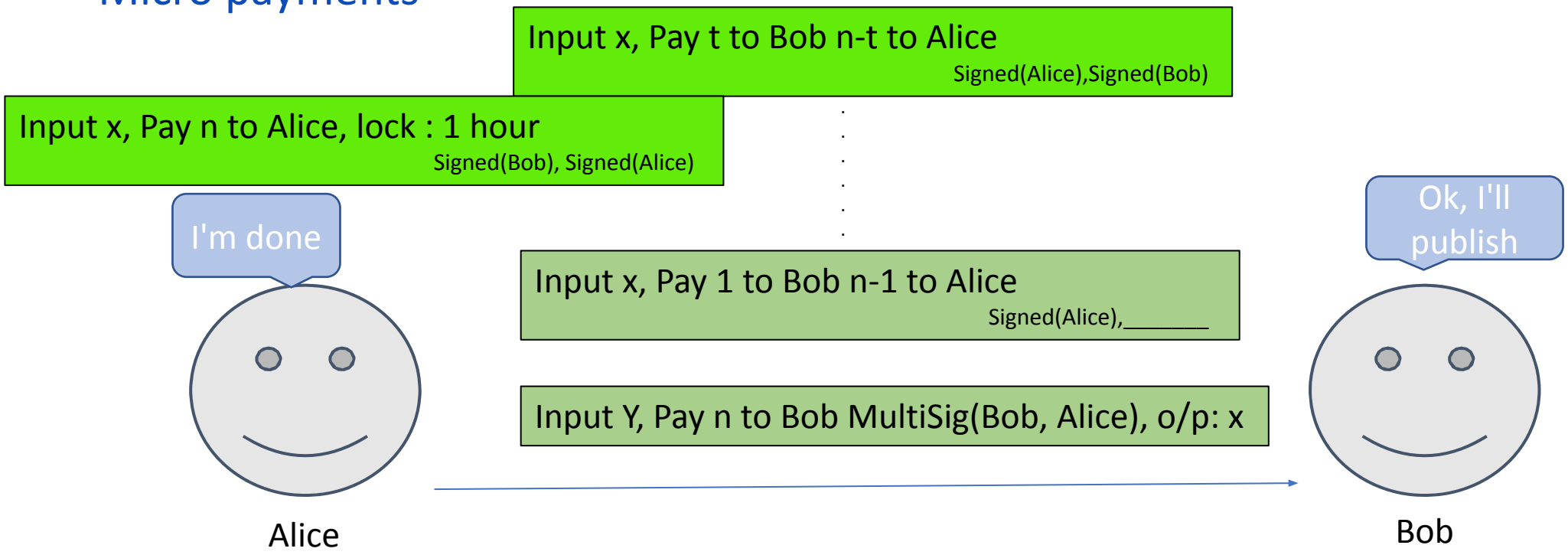
Micro payments



- Alice demands a refund transaction from Bob
- Alice broadcasts both refund and escrow transaction once she gets the refund transaction signed from Bob

Transactions in Bitcoin

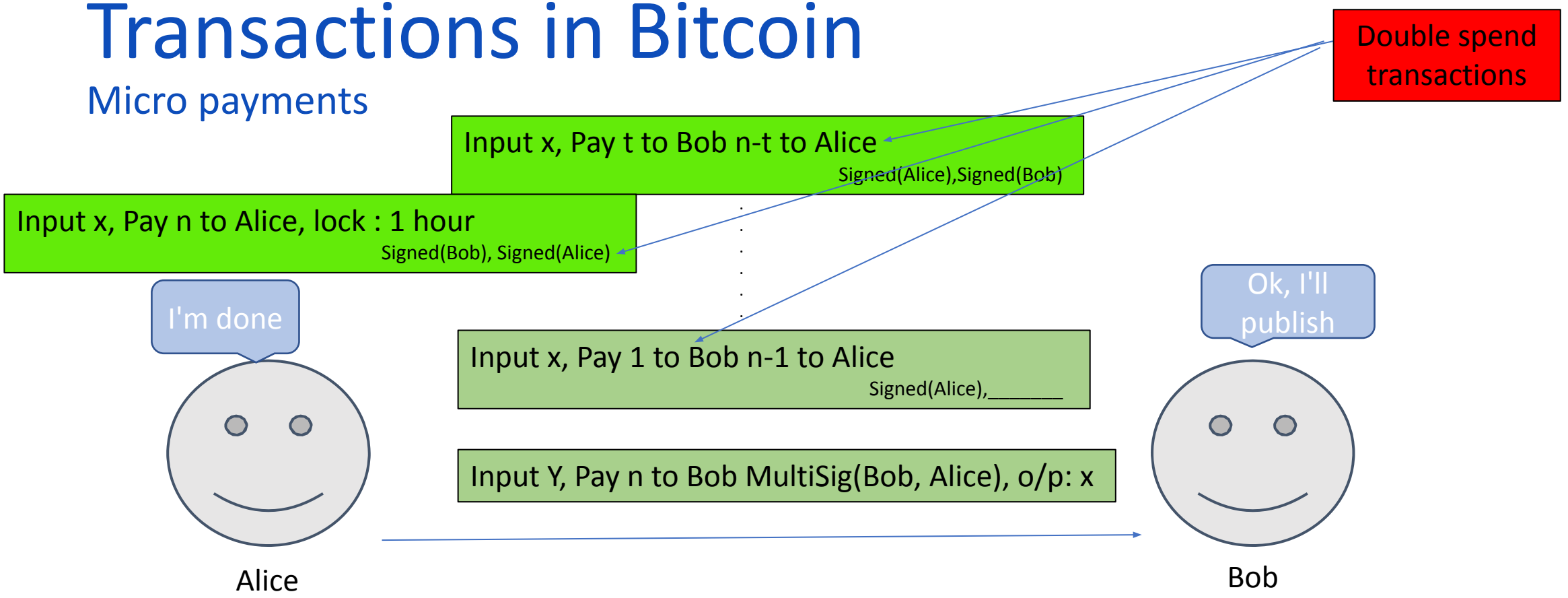
Micro payments



- If Bob doesn't sign
  - Alice will get her money back after 1 hour

Transactions in Bitcoin

Micro payments



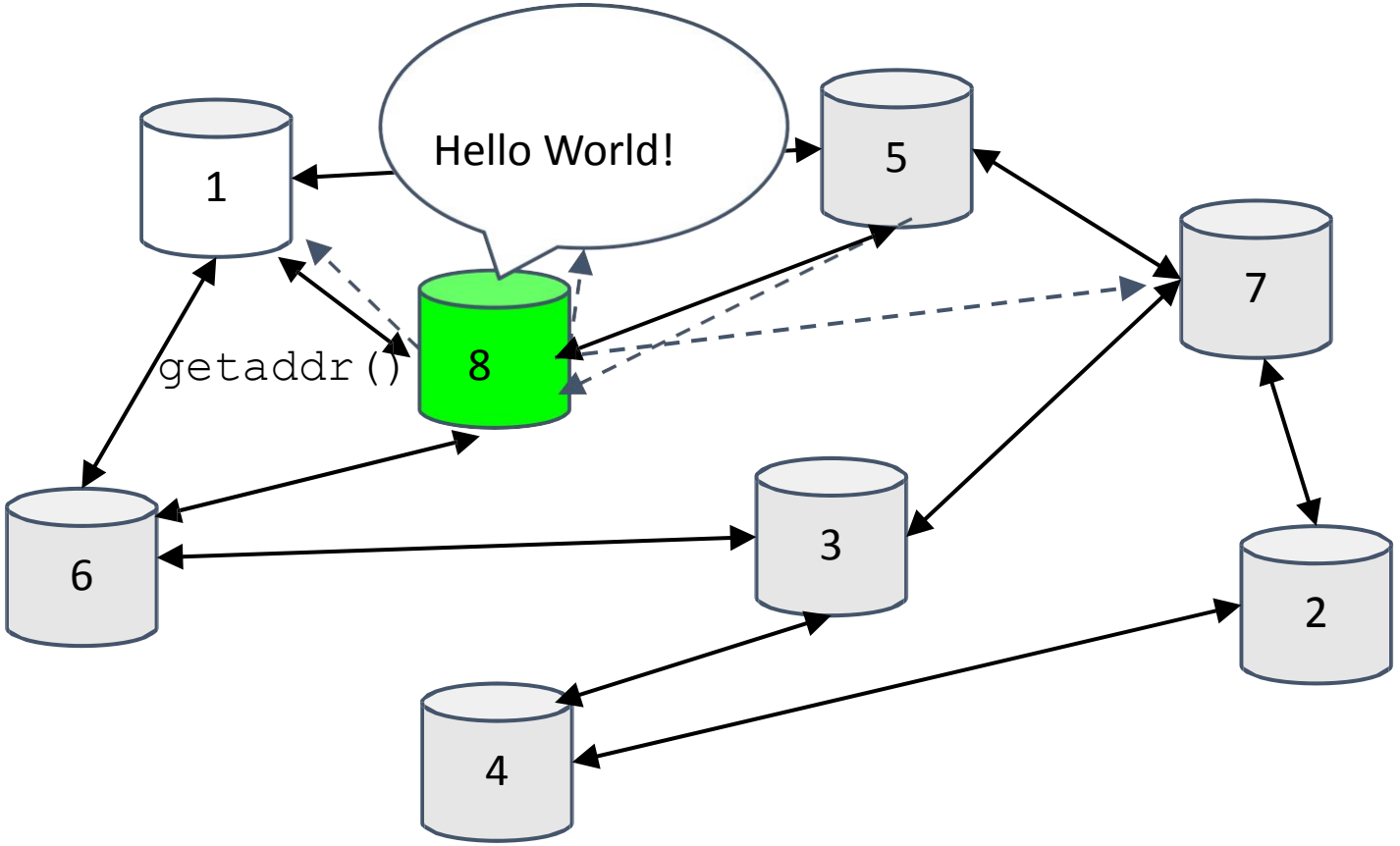
- What if Alice tries to redeem the refund even Bob signs the final transaction?
- HW : Analyse for different scenarios

P2P network

P2P network

- Ad-hoc protocol (runs on TCP port 8333)
- Ad-hoc network with random topology
- All nodes are equal
- New nodes can join at any time
- Forget non-responding nodes after 3 hr

Joining the network



- A new node connects to its nearest node
  - Seed node
  - Bitcoin has API for this

Joining the network

- Asks address of neighbouring nodes
- Can ask the address of their neighbouring nodes
- The process repeats for some time
- After this the nodes can choose the nodes to connect

Network

Message passing

- Gossip/flooding protocol
- Nodes run a series of checks before relaying a transaction

1. Transaction valid with current block chain
    - Ensures the unlocking script returns true
2. (default) script matches a whitelist
3. No double spend
4. Not an "already seen" transaction (why?)

- Sanity checks to keep the network robust
- Dishonest nodes can ignore them
- Note : These are transaction validity check not blocks
    - Lightweight

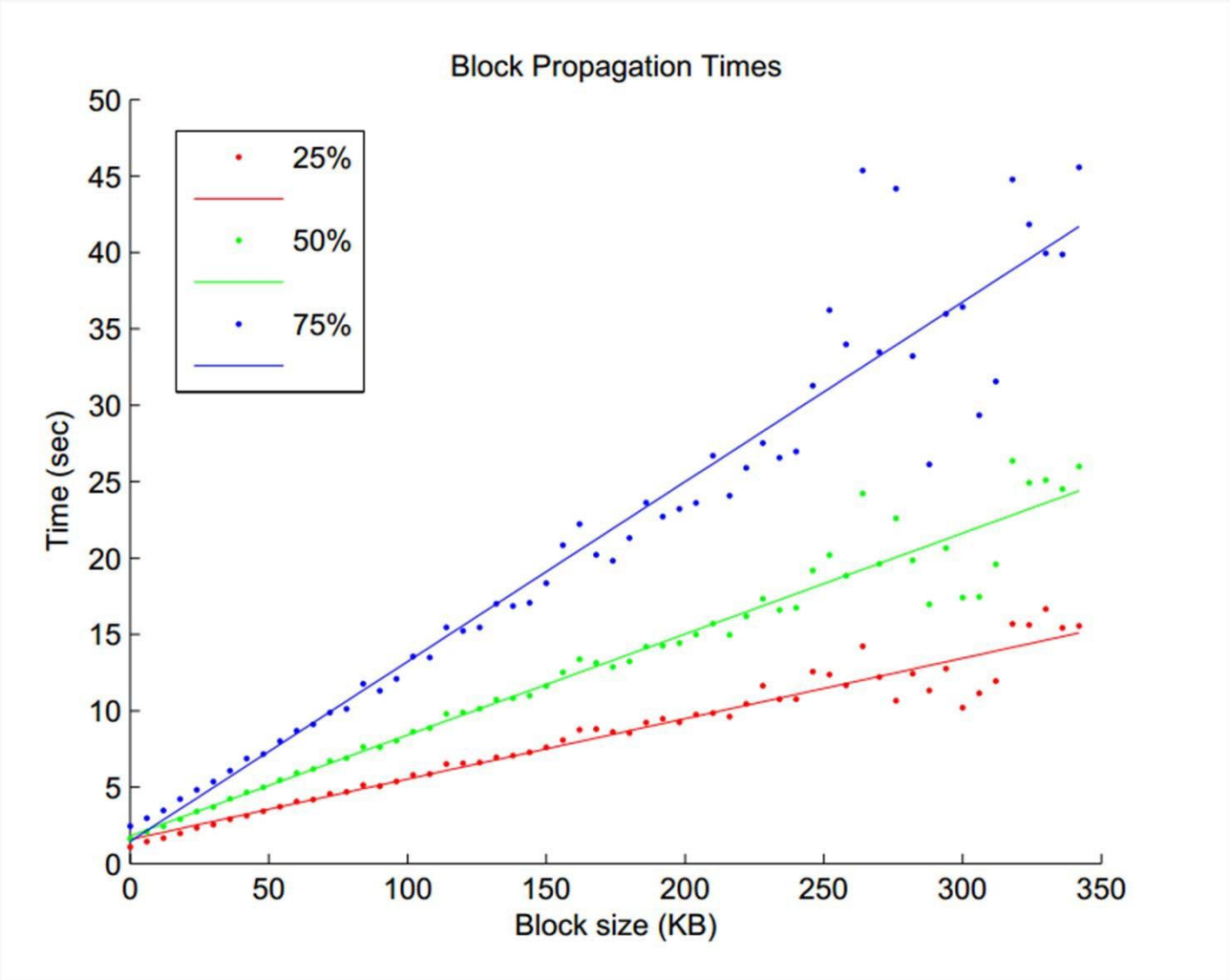
Network

Message passing

- Block passing is more complex
- Check for validity, no double spending
- Build on longest valid chain
- Some nodes might implement different logic
- Will create divergence/fork
  - The network can withstand this

Network

Propagation delay

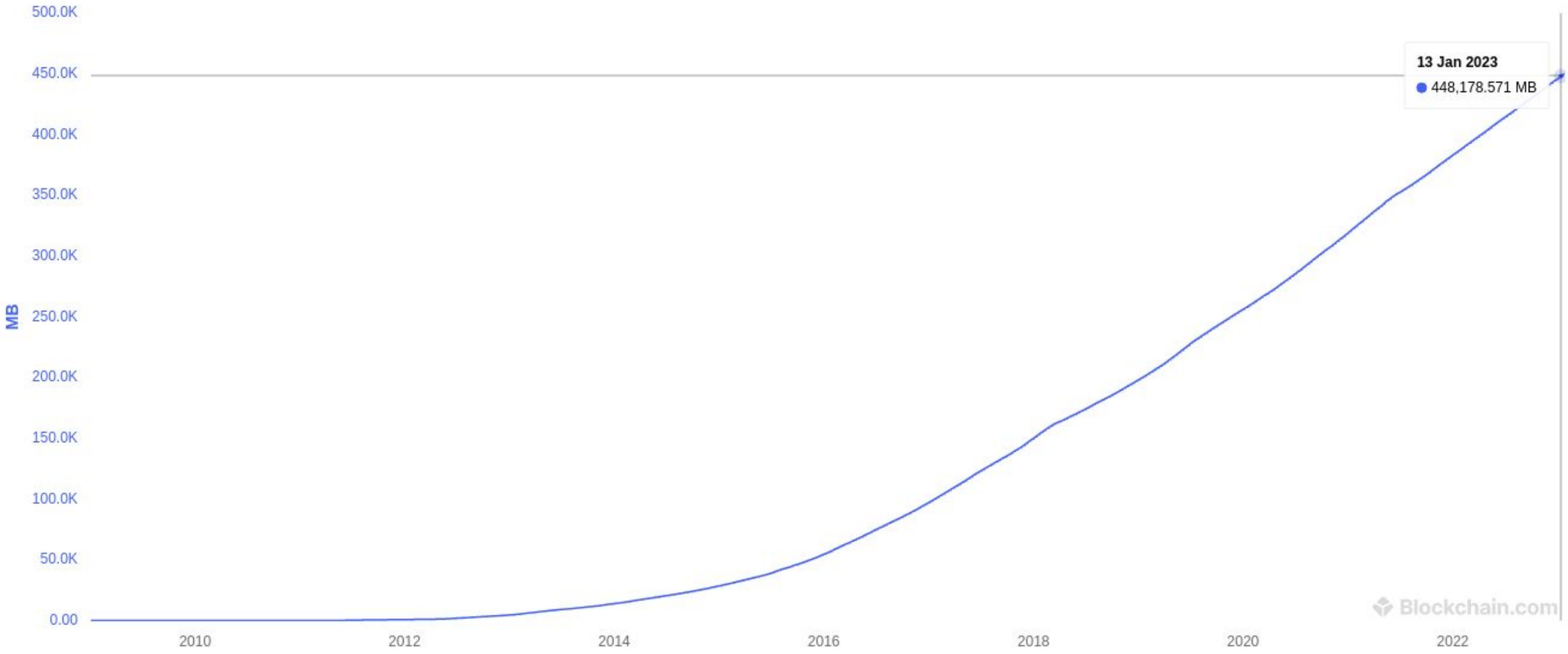


Source: Yonatan Sompolinsky and Aviv Zohar: “Accelerating Bitcoin’s Transaction Processing” 2014

Race conditions

- Transactions or blocks may conflict
- This is called “race condition”
- Nodes accept the transaction or block what they hear first
- Next miner decides what to include/exclude
- Network position matters
- Miners may implement other logic!

Size of the network



- Close to 500 gb

Size of the network

- Fully validating nodes
  - Permanently connected
  - Store entire blockchain
  - Hear and forward every node/transaction
- Thin/simple-payment verification (SPV) client
  - Idea: don’t store everything
  - Store block headers only
  - Request transactions as needed
  - To verify incoming payment
  - Trust fully-validating nodes
  - 1000x cost savings!
  - e.g., wallets

Size of the network

- Impossible to measure exactly
- Nodes connect and disconnect unpredictably
- Estimates-up to 1M IP addresses/month
- Only about 5-10k “full nodes”
  - Permanently connected
  - Fully-validate

The End !!!