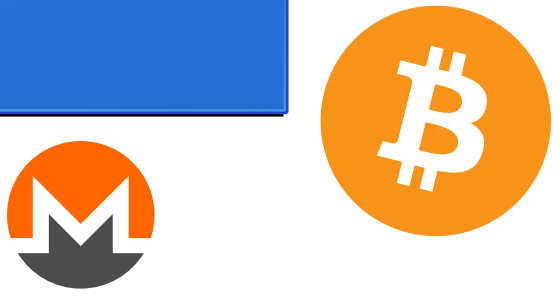
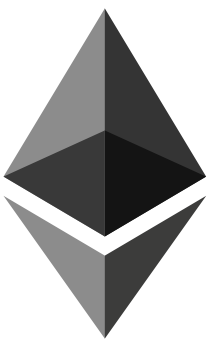
# *Gentle* Introduction to Blockchains

Orfeas Stefanos Thyfronitis Litos
PhD in Cryptography and Blockchains
University of Edinburgh
11/10/2019

# Outline

- Why it works
- Smart contracts
- Types, ideas and the future



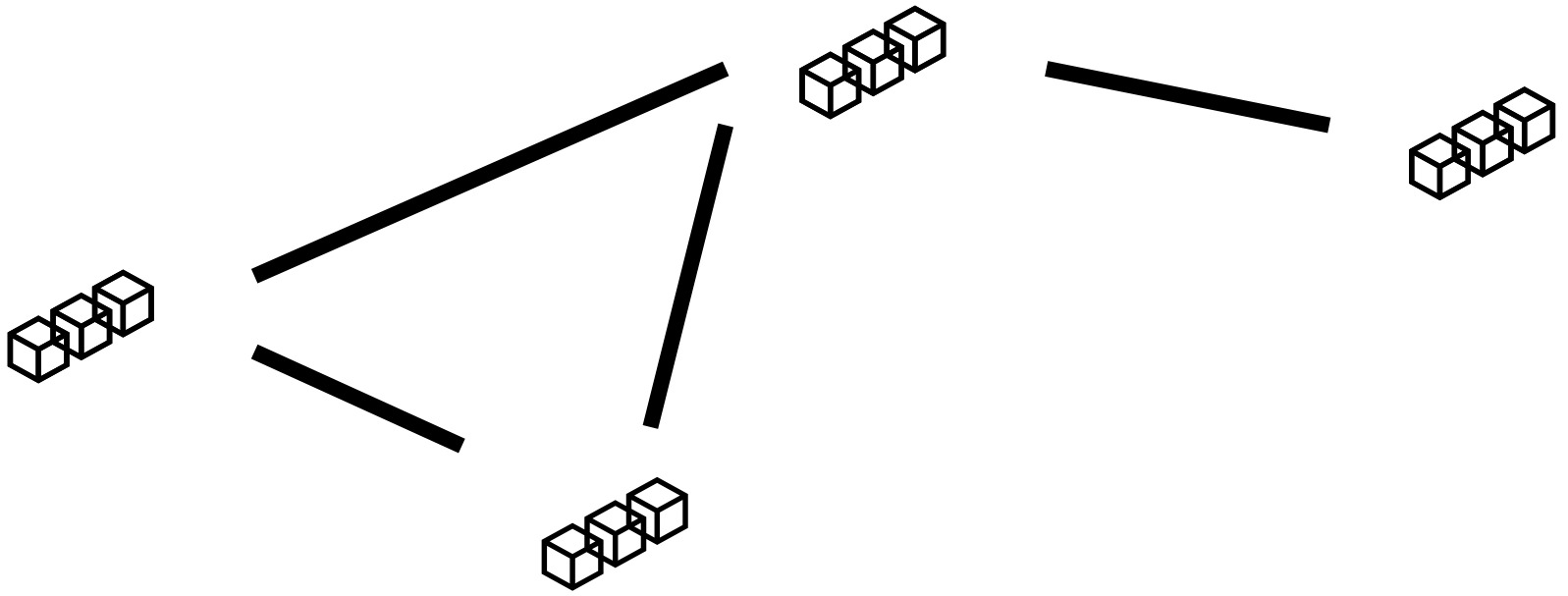

# Part I

Why it works, or a Bitcoin primer

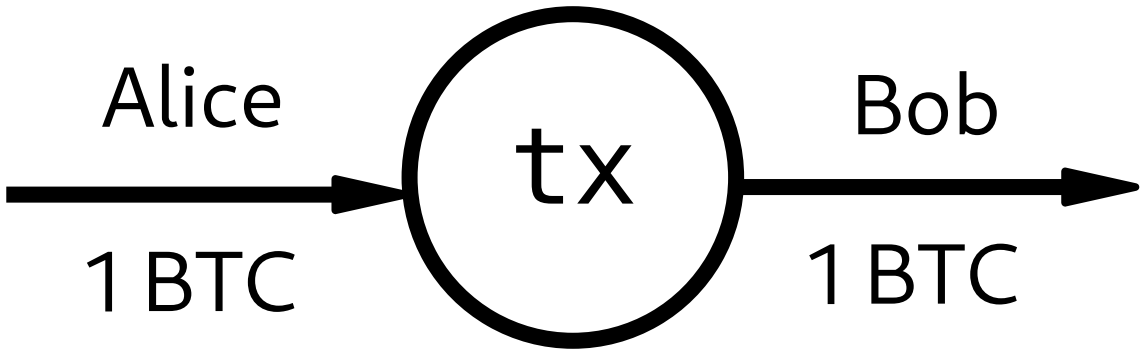
# What is a Blockchain?

- distributed,
- append-only,
- transaction ledger

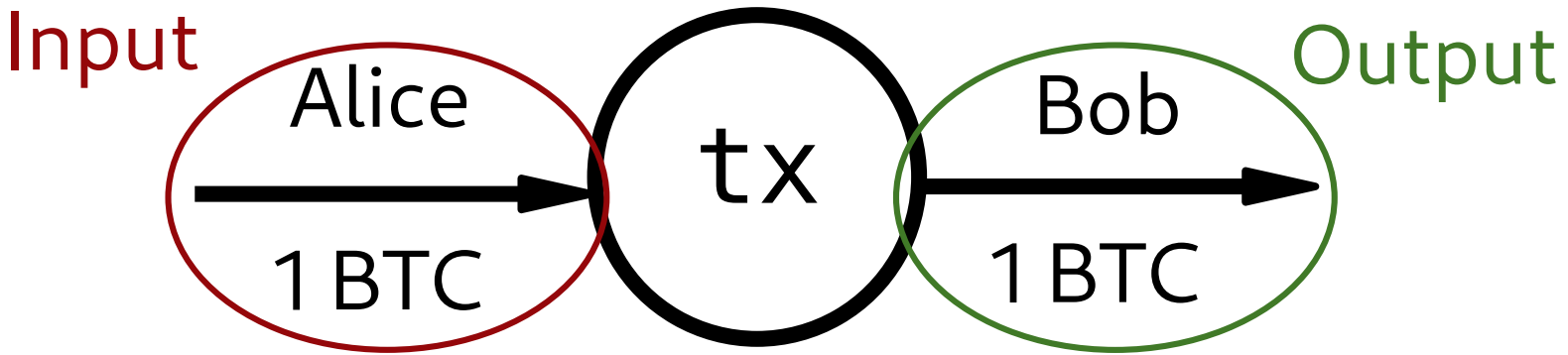
# What is a Blockchain?



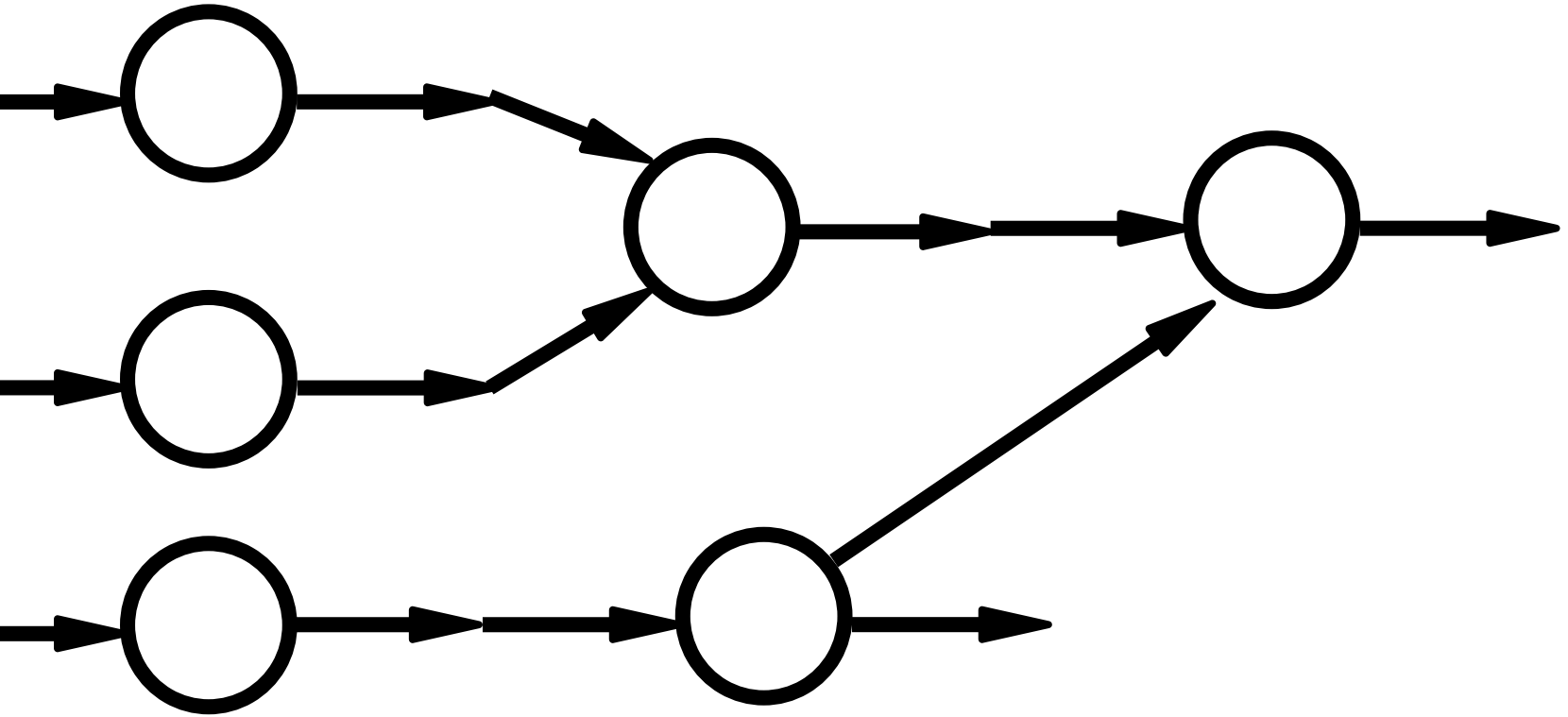
# Transactions



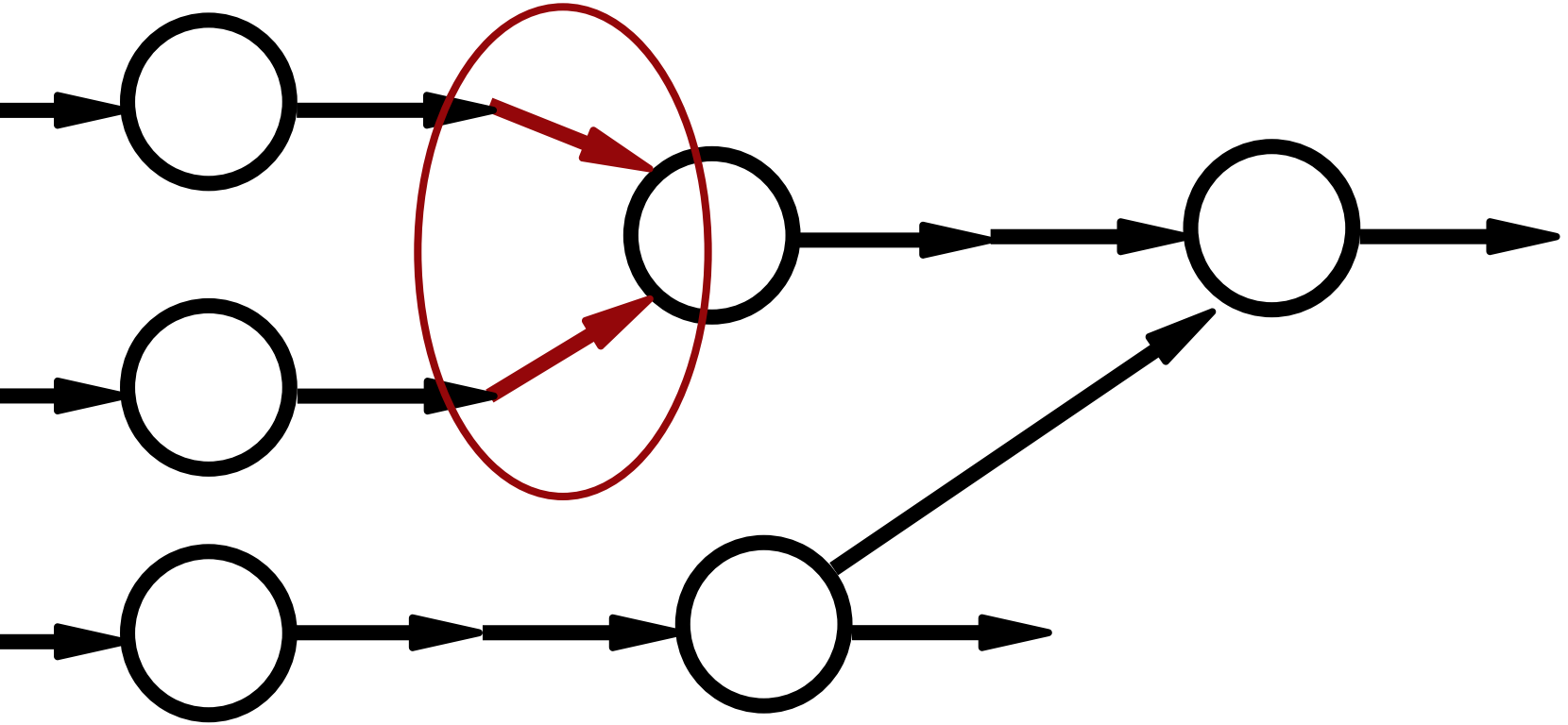
# Transactions



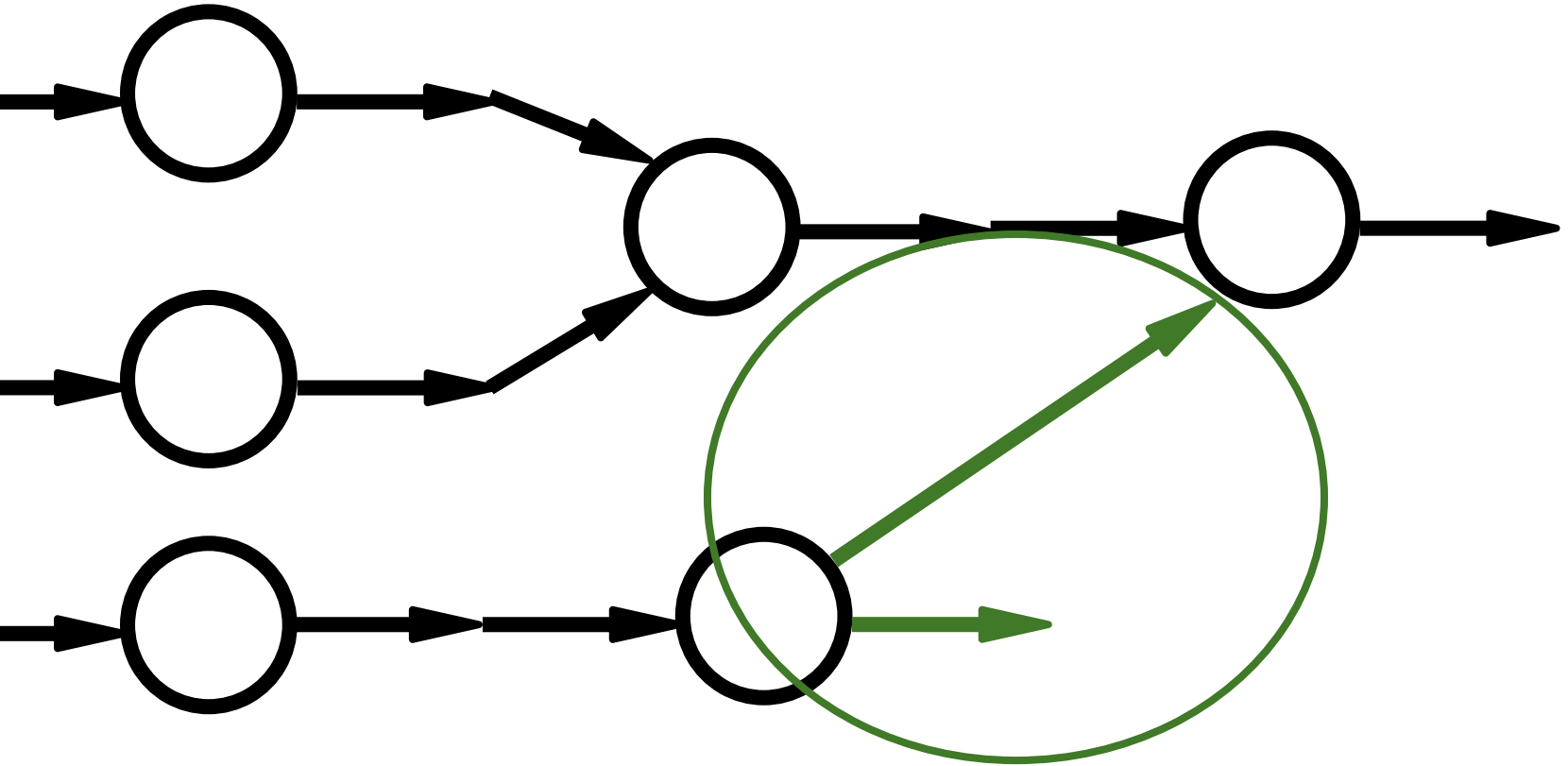
# Transaction Graph



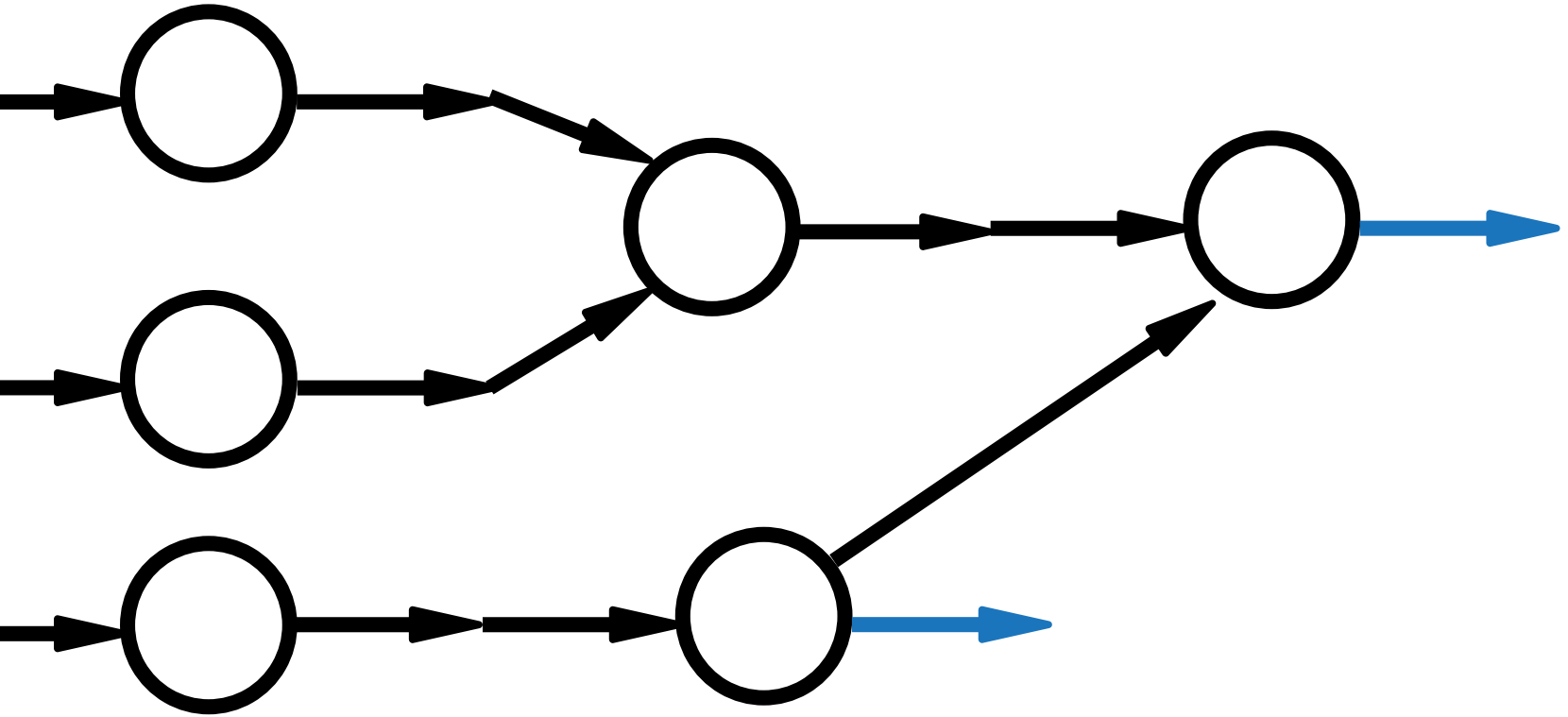
# Many inputs



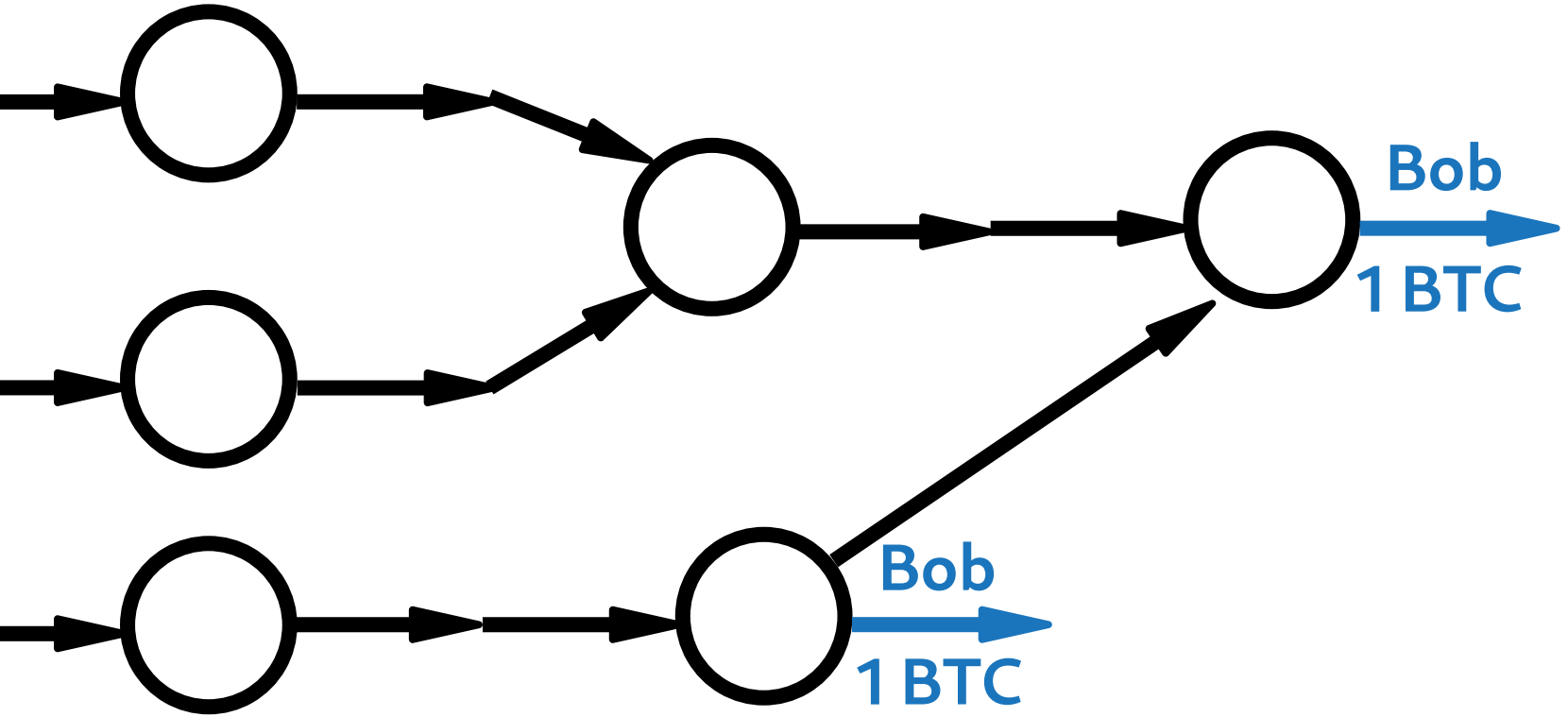
# Many outputs



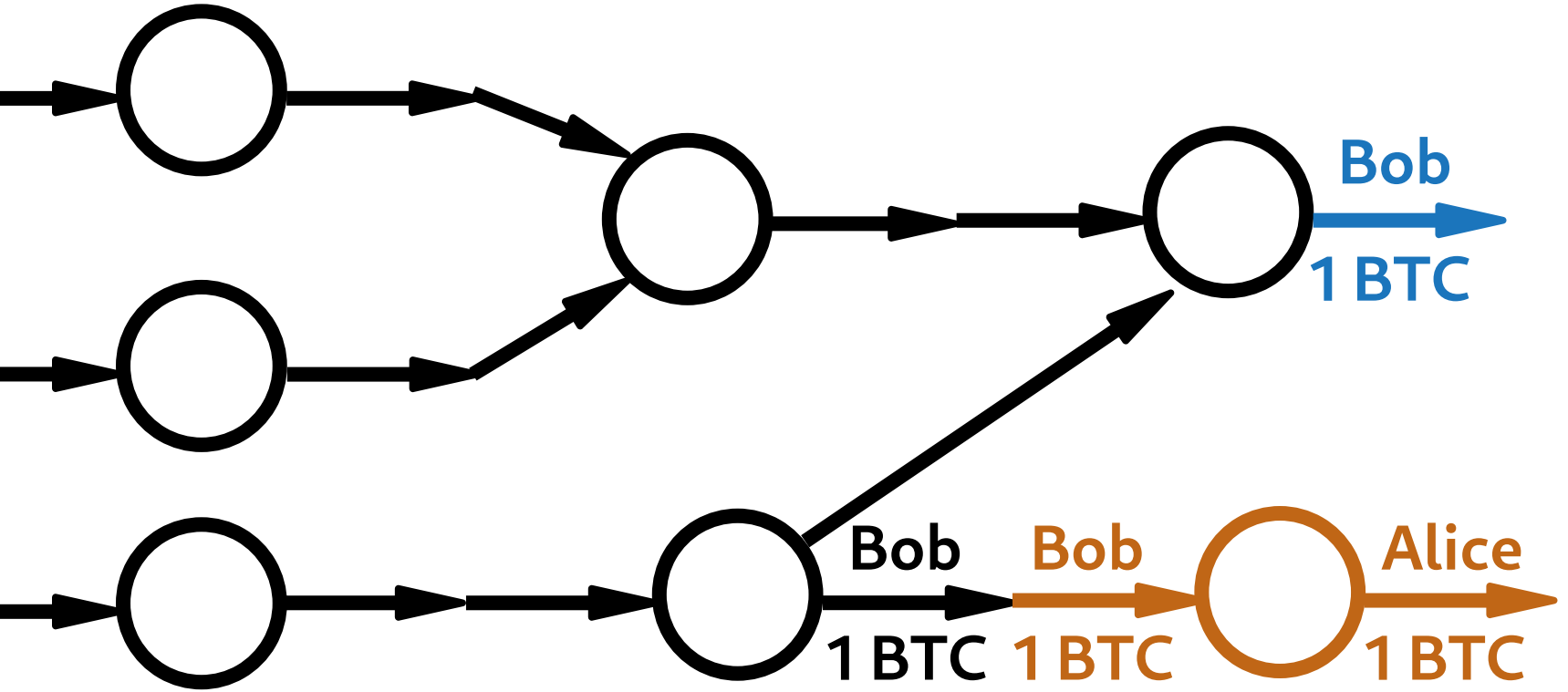
# Unspent Transaction Outputs (UTXO)



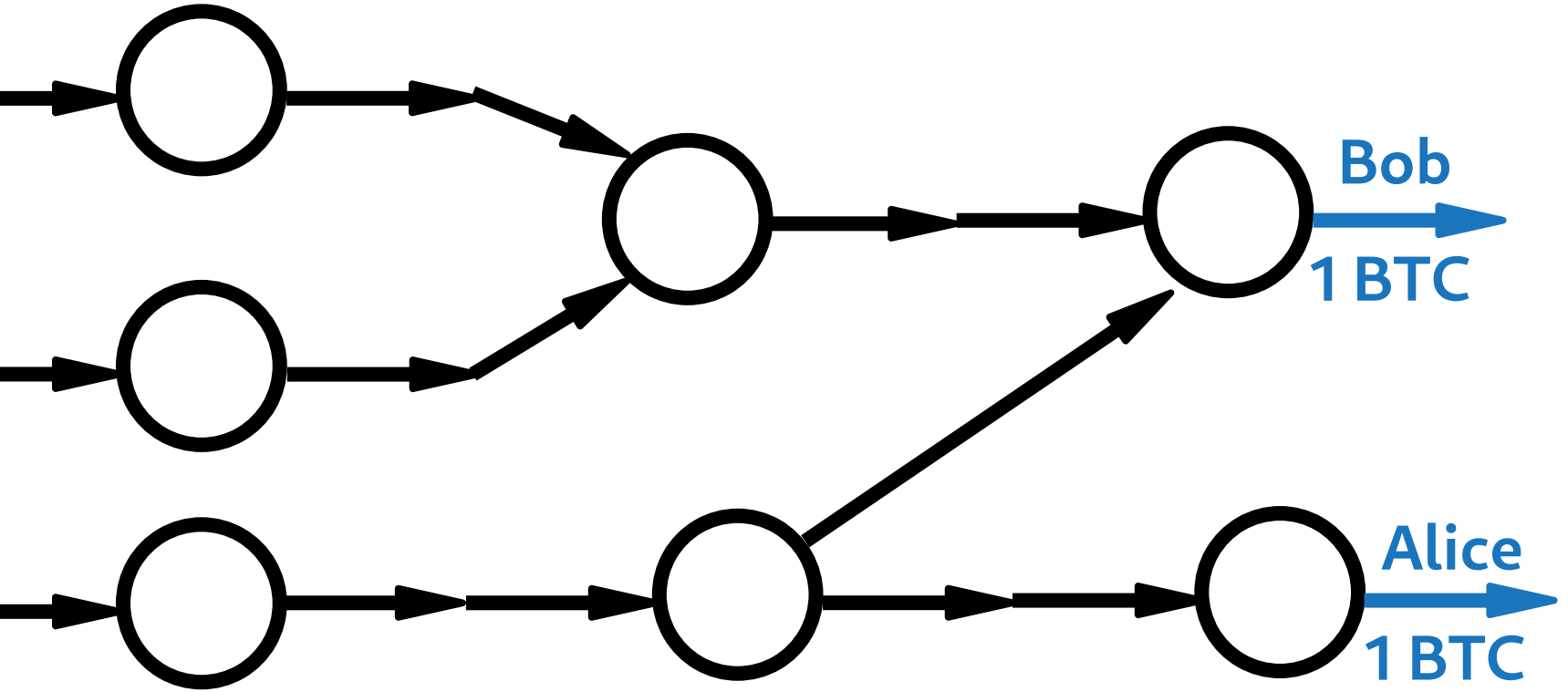
# Bob has 2 BTC



# Bob pays 1 BTC to Alice



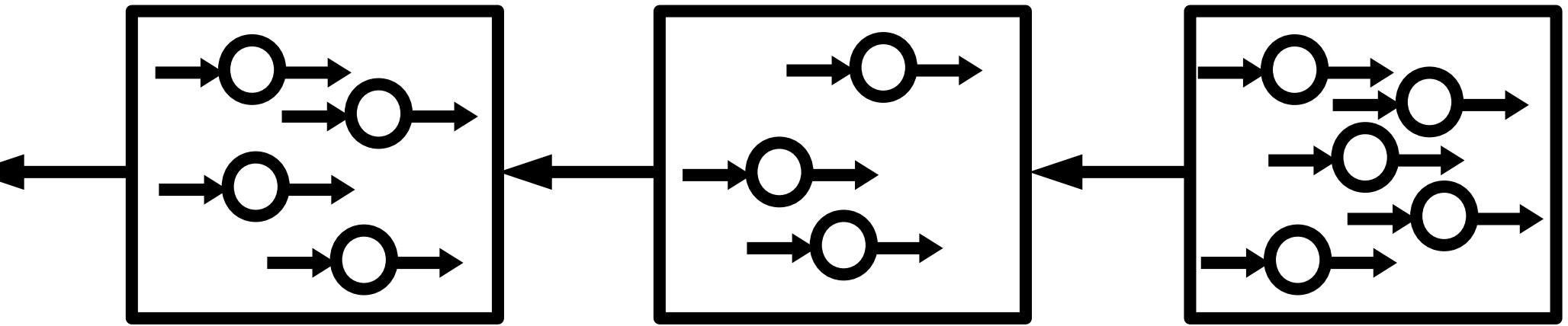
# New UTXO



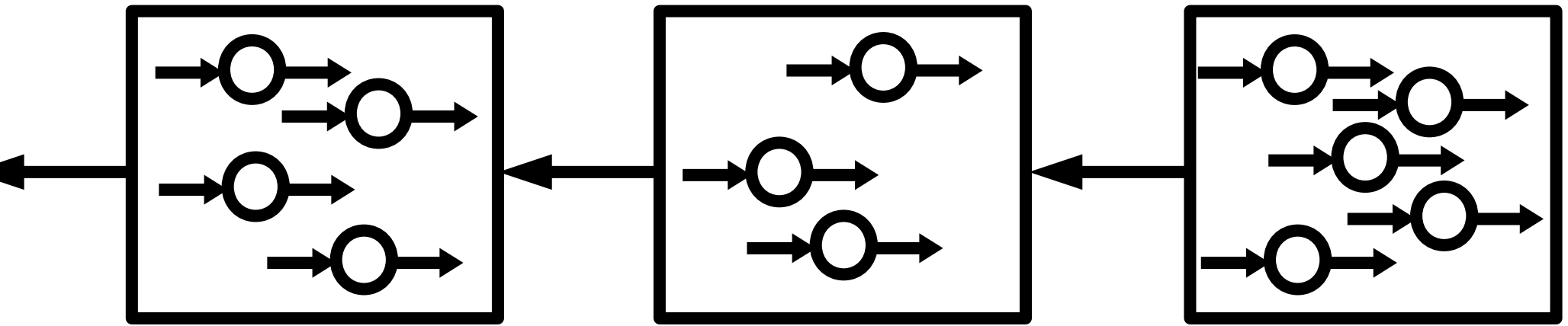
# Block

- Contains
  - transactions
  - metadata
- Has unique parent

# Blockchain

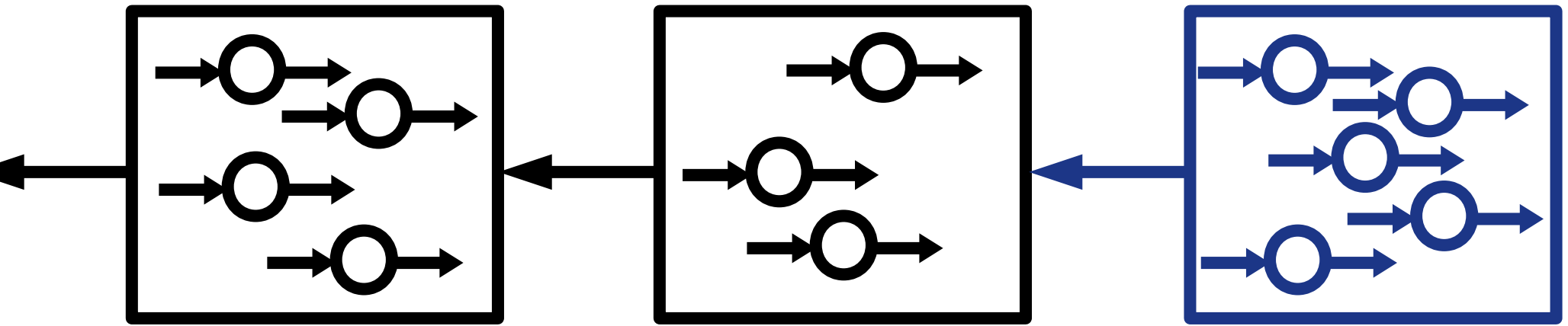


# Blockchain

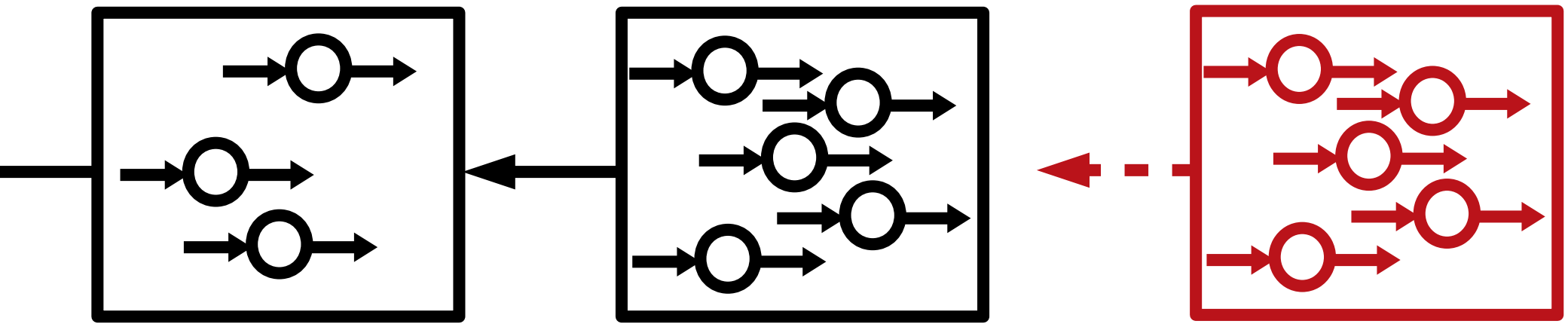


History of all transactions

# Blockchain

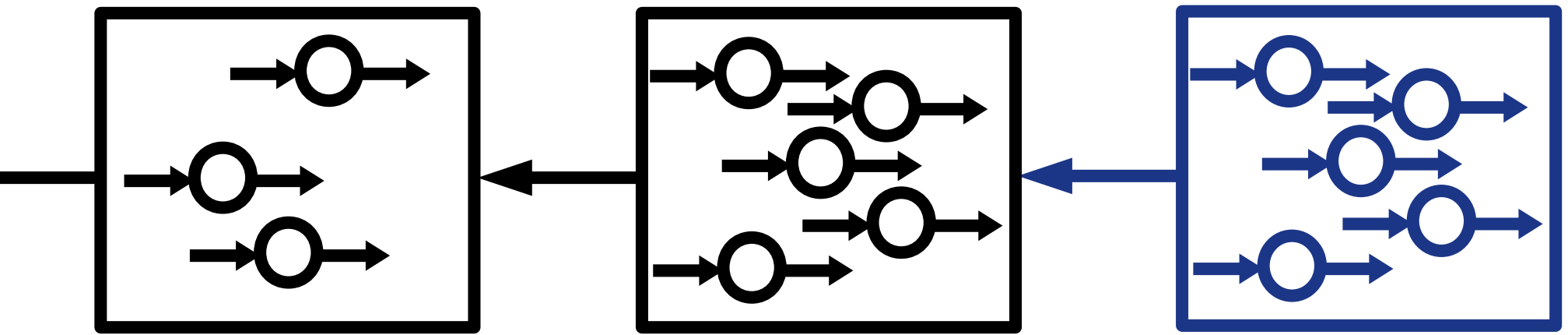


# Mining (a.k.a. writing history)



- Only valid transactions?
- Proof of Work?

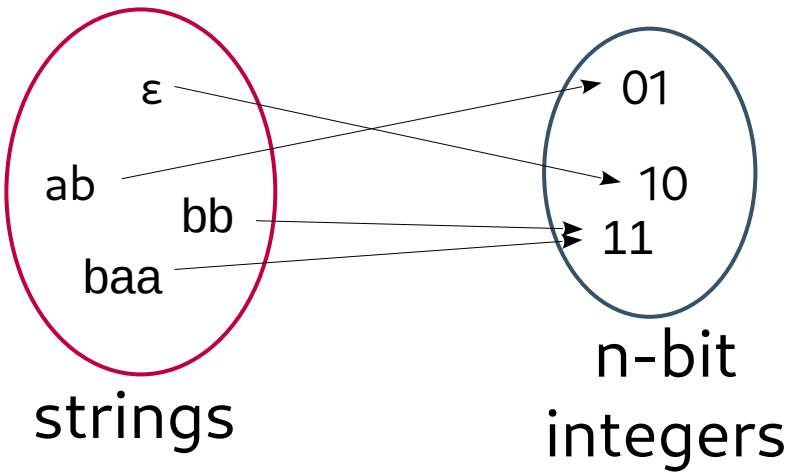
# Mining (a.k.a. writing history)



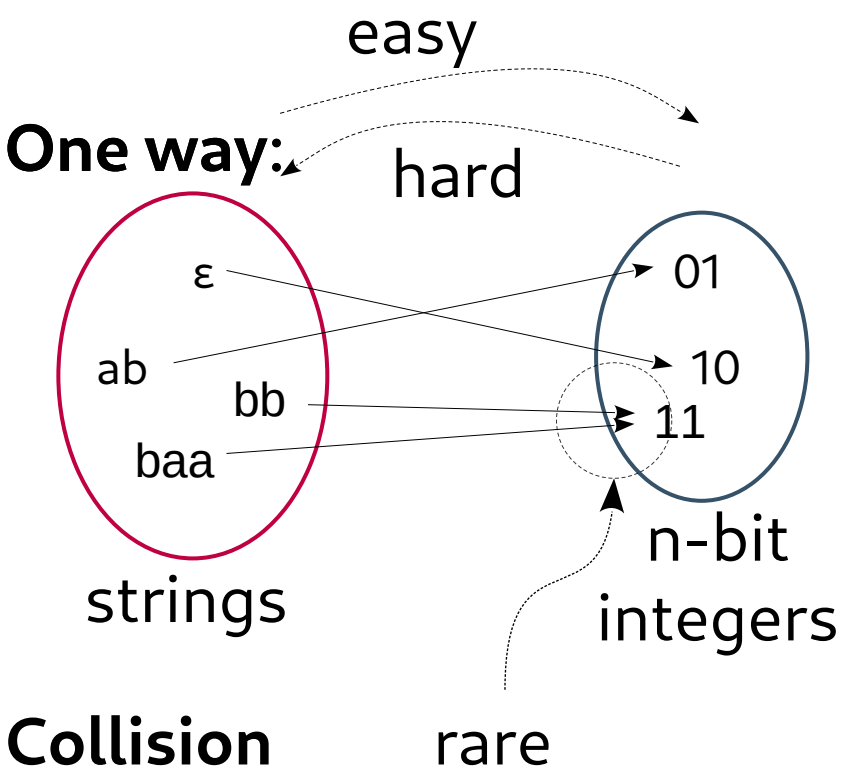
- Only valid transactions!
- Proof of Work!

**new tip**

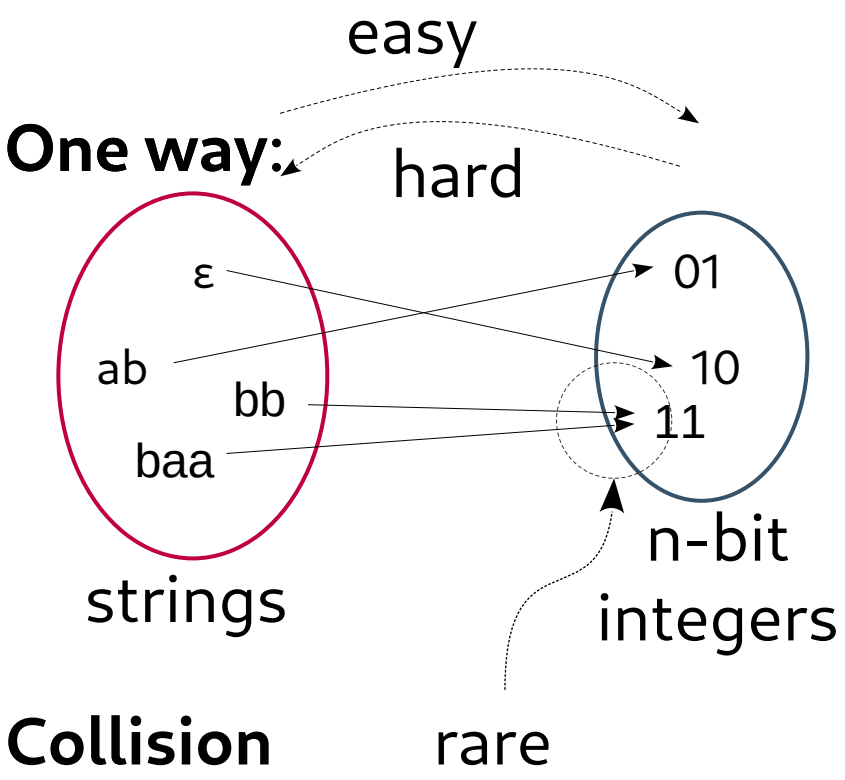
# Hash function



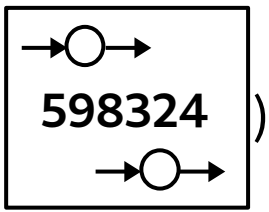
# Hash function



# Hash function




Bitcoin uses SHA256, e.g.:

SHA256(  ) =



000000000000000000
d7819fc59c65ca6f3e8f73c
6eeadf538aa874a31341fb

# Proof of work

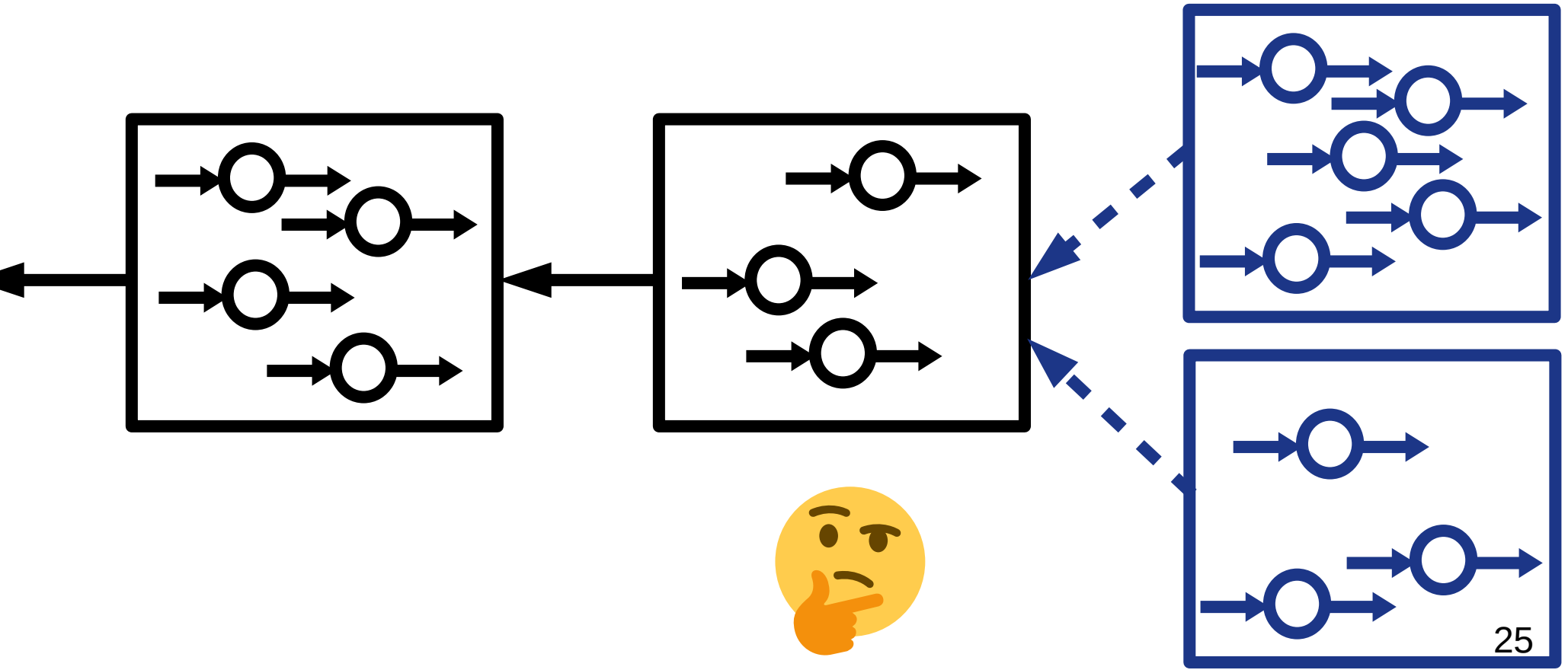
```
def hasProofOfWork(block):
  return hasManyLeadingZeroes(SHA256(block))
```

Valid blocks need energy

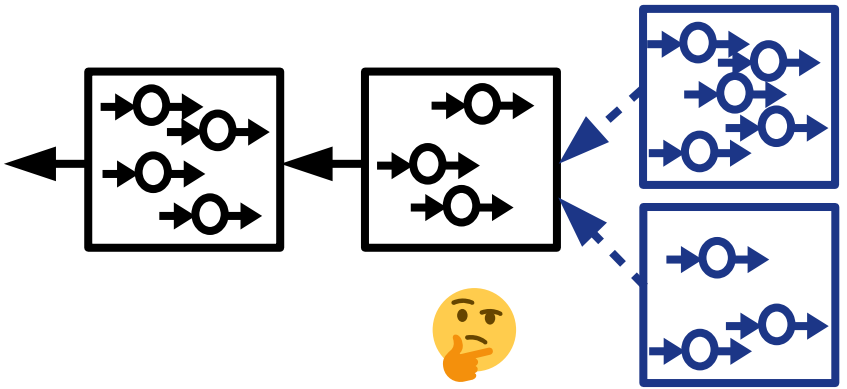
→

can’t spam blocks

# Forks



# Forks



Protocol says:

- Choose longest fork
- If equal, choose random

# Nice properties

**if** *honest* mining power > 50%,

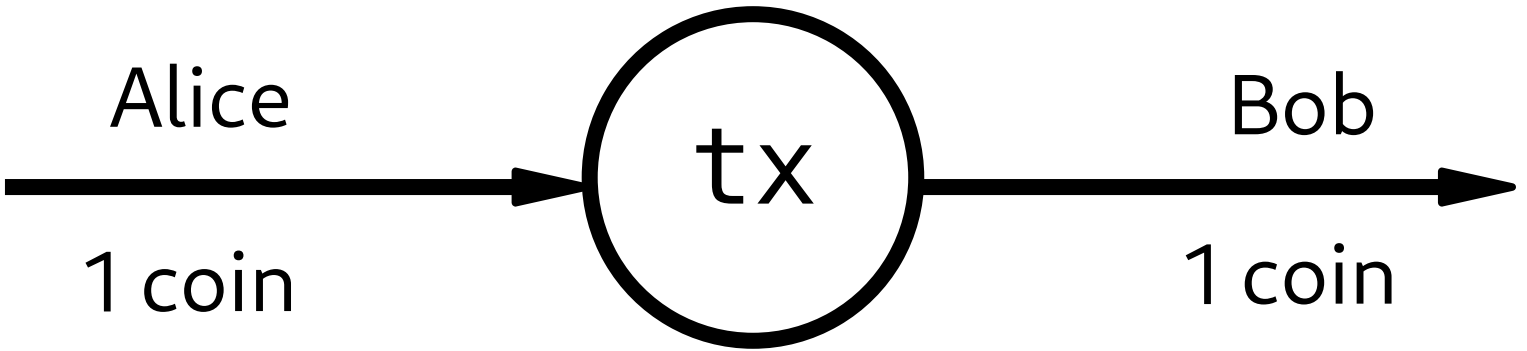
- **Liveness**: a new tx *will enter* the chain
- **Persistence**: Old blocks *won't change*

Garay, Kiayias, Leonardos. "The bitcoin backbone protocol: Analysis and applications." Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2015.

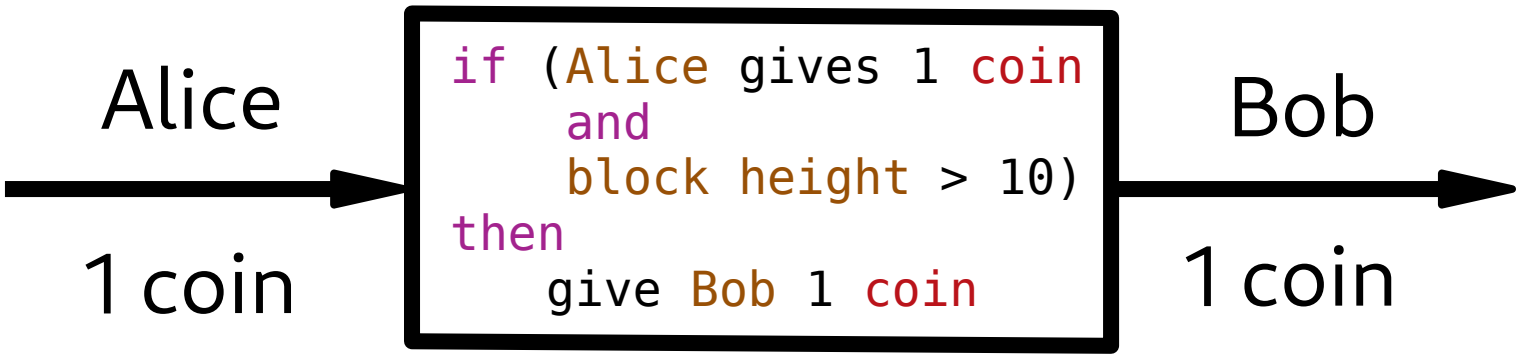
# Part II

## Smart Contracts: Programmable money

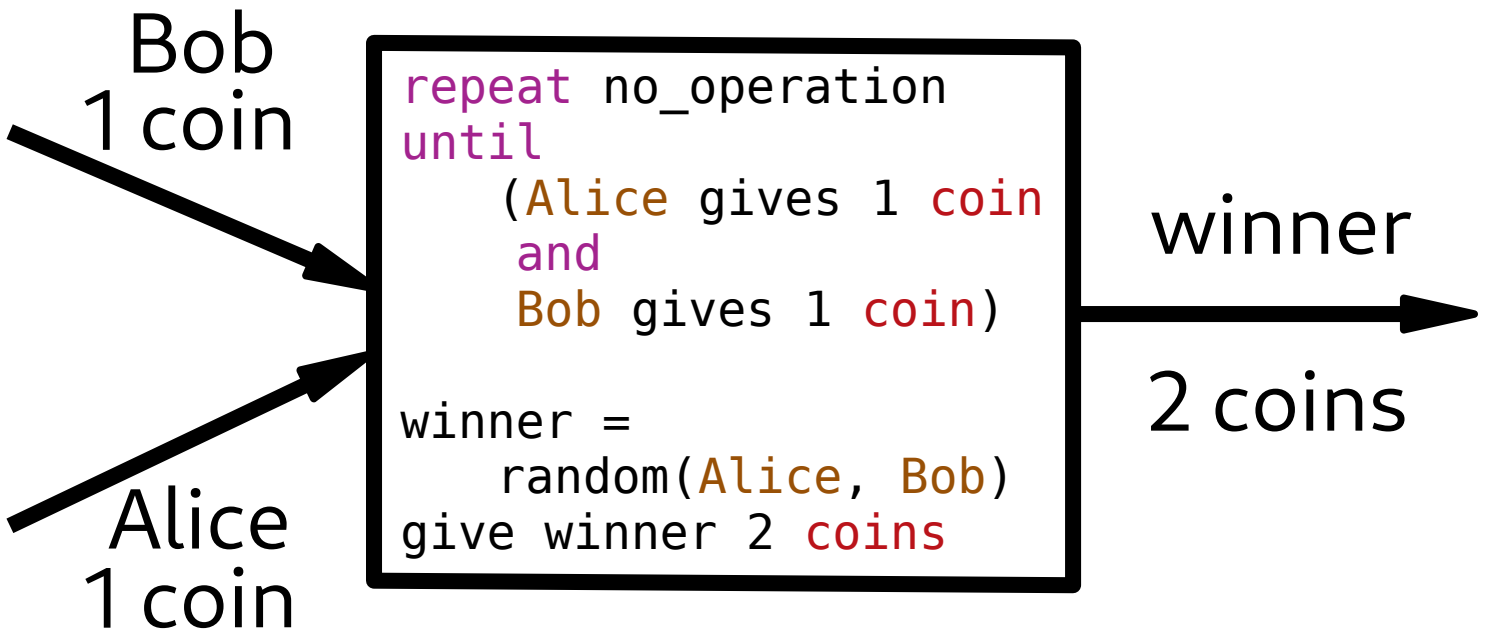
# Remember transactions?



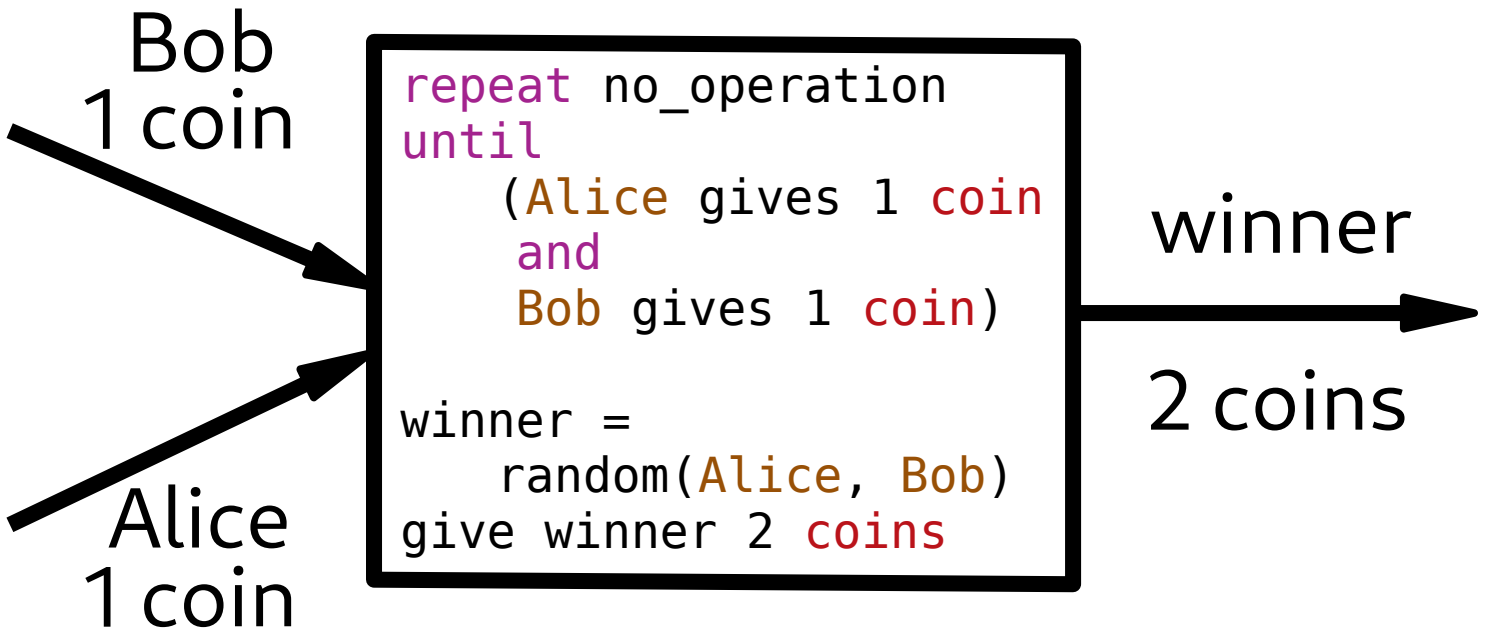
# Now add some code!



# Example 1: Flip a coin



# Example 1: Flip a coin




**Don’t use this contract in real life!**

# Example 2: King of the Hill

```
top = 0
king = null
while (true)
  if (user gives x coins
      and
      x > top)
     give king top coins
     king = user
     top = x
```

**Don’t use this contract in real life!**

# Still too hard

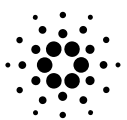
A fast way to lose your (and others')

Smart Contracts: ~~Programmable~~ money

- Very easy to make mistakes
  - TheDAO
  - Parity wallet
- Smart contract languages not (yet) safe

# Part III

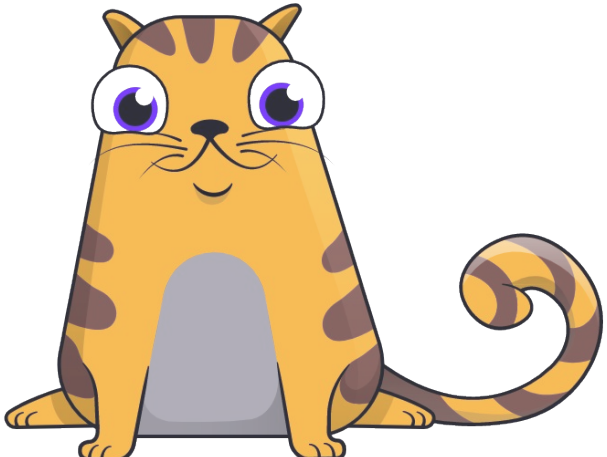
Types of blockchains, applications and the future



Xchain

# Cryptokitties

Cute kitties that live on Ethereum!



# MakerDao

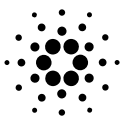
- Decentralised stablecoin
  - On Ethereum
- 1 Dai = 1 USD
- Overcollateralised by floating “Maker”

# Types of blockchains

- Permissioned vs Permissionless


- Proof of Work vs Proof of Stake


v



- Private vs Transparent

# Scalability issue


: 7 txs/sec

VISA: 20,000 txs/sec

Problem: too much redundancy

# Candidate Solution

Payment Channels! (e.g. Lightning, LND)

- 1 tx on-chain to open channel
- Unlimited off-chain txs
- 1 tx on-chain to close channel

# Crosschain transactions

Move coins to another chain

E.g. use bitcoins in Ethereum contracts

# Interconnected blockchains

Specialised blockchains

Separation of duties

Related: Sharding

# Questions?


Credits:

- Blockchain by Pablo Rozenberg from the Noun Project