



# **Blockchain for Public Administrations**

## **Part I: Bitcoin and beyond**

**Massimo Bartoletti**

University of Cagliari  
[blockchain.unica.it](http://blockchain.unica.it)





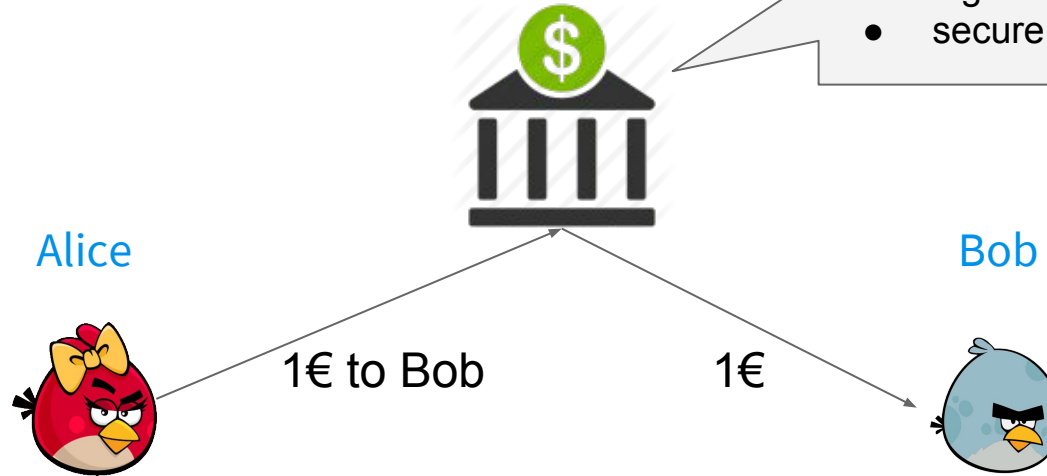
## Plan of the tutorial

1. The archetypal blockchain: Bitcoin
2. Bitcoin problems
3. Post-Bitcoin blockchains
4. Blockchain for Public Administrations  
(Prof. Andrea Vitaletti, Univ. Roma “La Sapienza”)



# The archetypal blockchain: Bitcoin

## Payments with banks



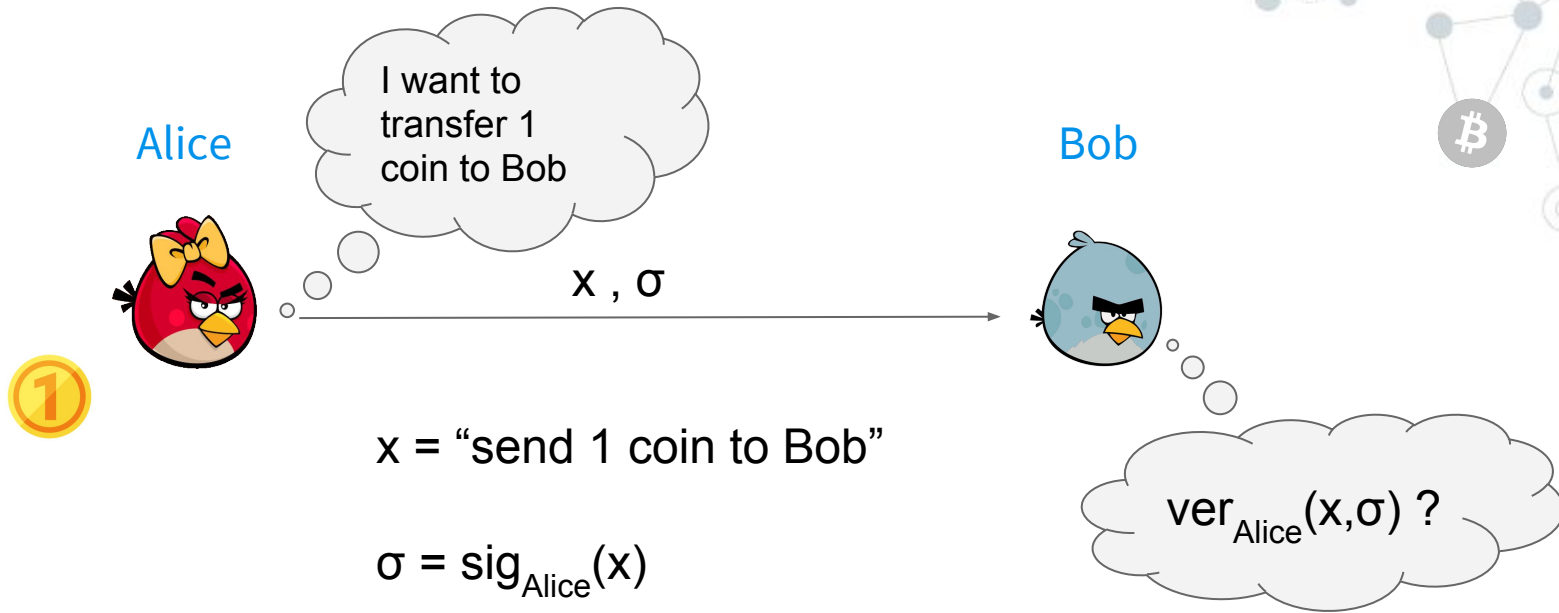
Bank is a **trusted authority** (can drop transactions, steal money, ...)

Despite the centralization, **anonymous** payments are possible:

D. Chaum. Blind signatures for untraceable payments. CRYPTO, 1982

+ many other works on cryptography in the 1990s

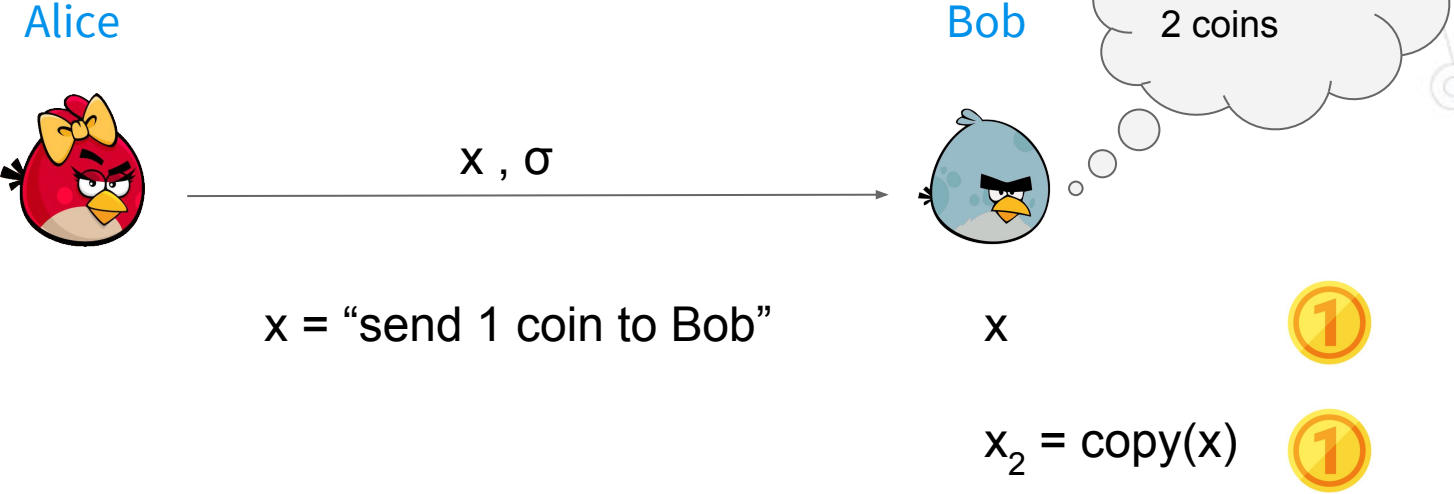
## Payments without banks: a naïve attempt



By verifying the signature  $\sigma$ , Bob can be sure about the authenticity of the message  $x$ : it's Alice who signed it!

**What is the problem?**

# Payments without banks: a naïve attempt



## Problem

If Bob is dishonest, he can **forge** coins !

# Payments without banks: using a public ledger

Alice



I want to transfer 1 coin to Bob

Bob



T0
in: -
wit: -
out(x): $ver_{Alice}(x)$
val: 1 coin

The **transaction** T0 certifies that Alice owns 1 coin.

This coin can be transferred to another transaction, that provides a witness satisfying the predicate out(x)

# Payments without banks: using a public ledger

Alice



Bob



Now I own  
1 coin

Cryptographic  
hash of the  
predecessor

T0
in: -
wit: -
out(x): $ver_{Alice}(x)$
val: 1 coin

T1
in: T0
wit: $sig_{Alice}(T1)$
out(x): $ver_{Bob}(x)$
val: 1 coin

**Bob cannot forge coins! He can only transfer his coin**



# The Bitcoin blockchain


**Blockchain** =  
sequence of  
transactions (grouped  
into blocks)

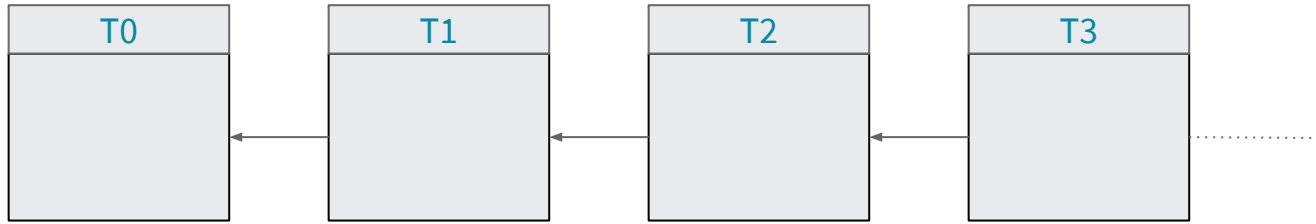
Alice



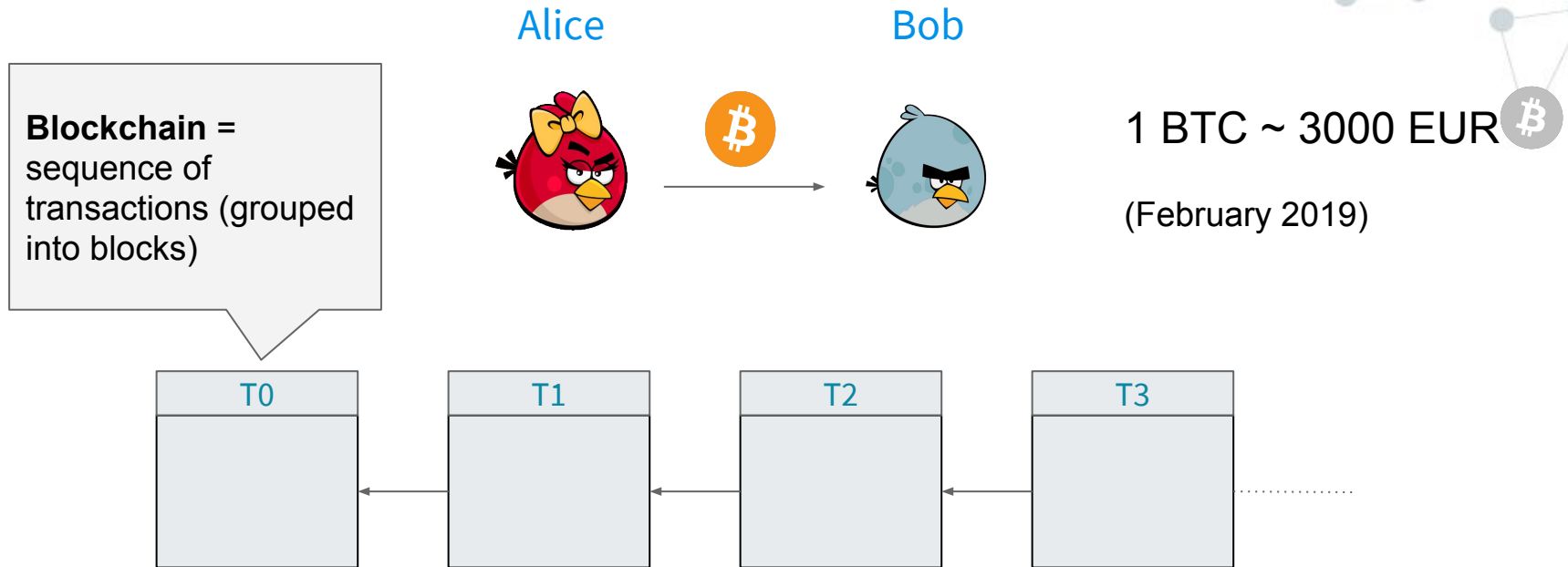
Bob



1 BTC ~ 3000 EUR   
(February 2019)



# The Bitcoin blockchain



The blockchain is:

- **permissionless**: anyone can add transactions
- **public**: anyone can read it (and compute the balance of each user)

**What is the problem?**

## The Bitcoin blockchain

**Problem #1** who owns the blockchain?

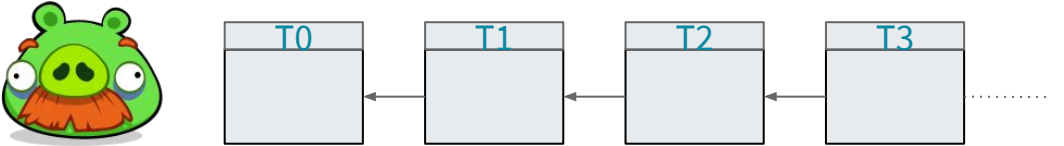
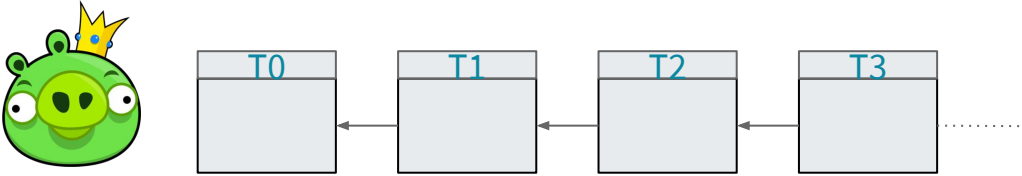
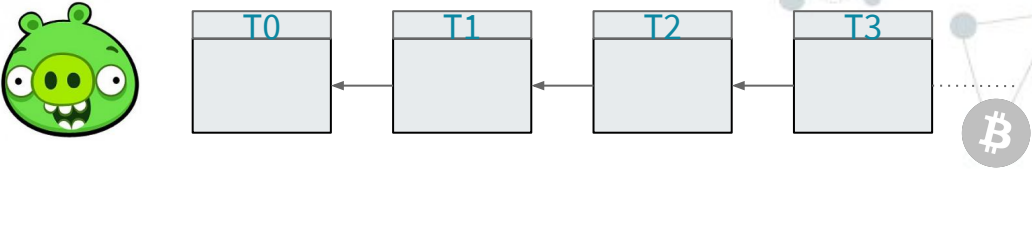
- ✗ central authority
- ✓ a peer-to-peer network (nodes do **not** trust each other)

**Problem #2** how is the blockchain updated?

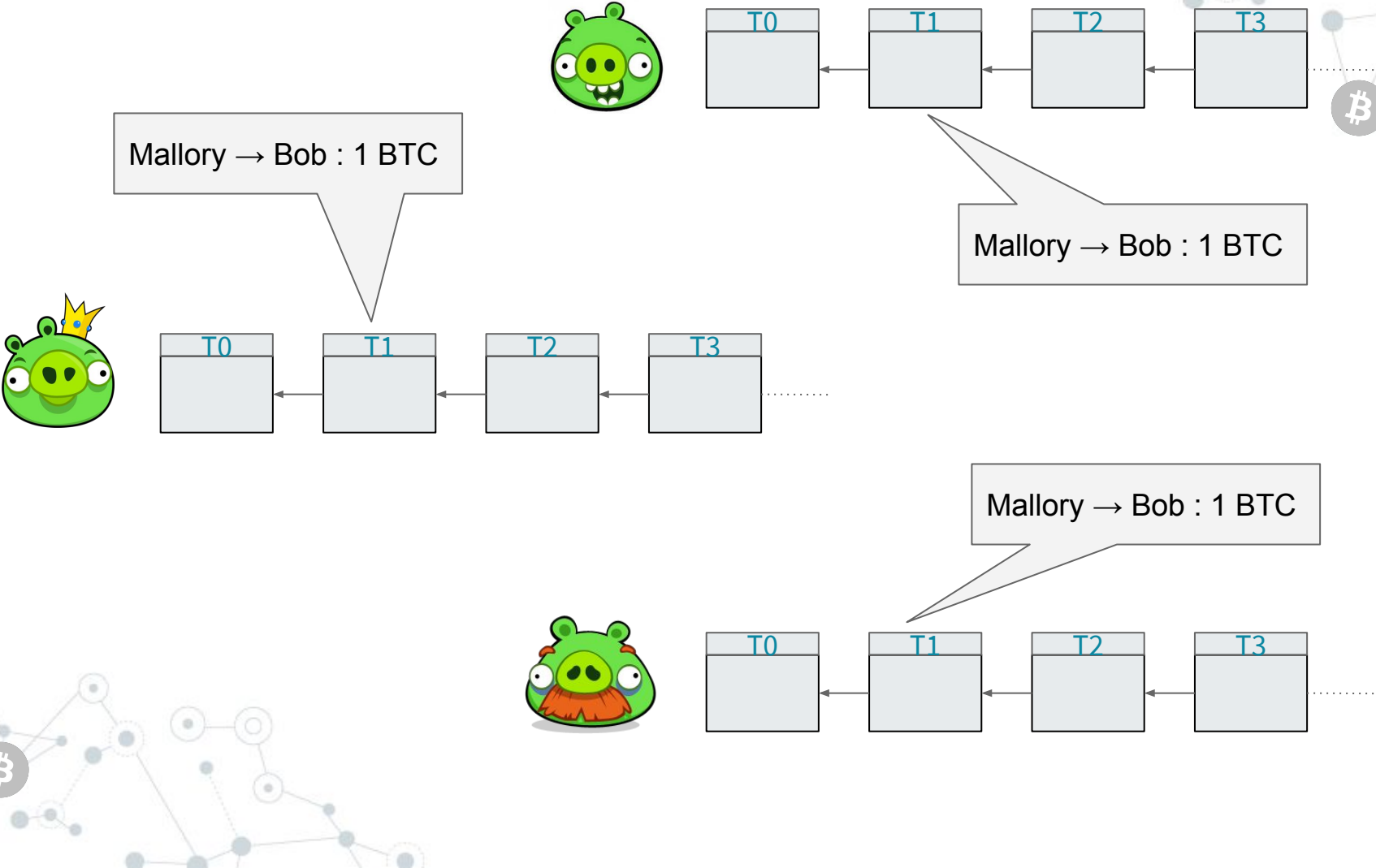
- ✗ remove / edit existing transactions
- ✓ only append transactions

**Problem #3** how can we guarantee **consistency**?

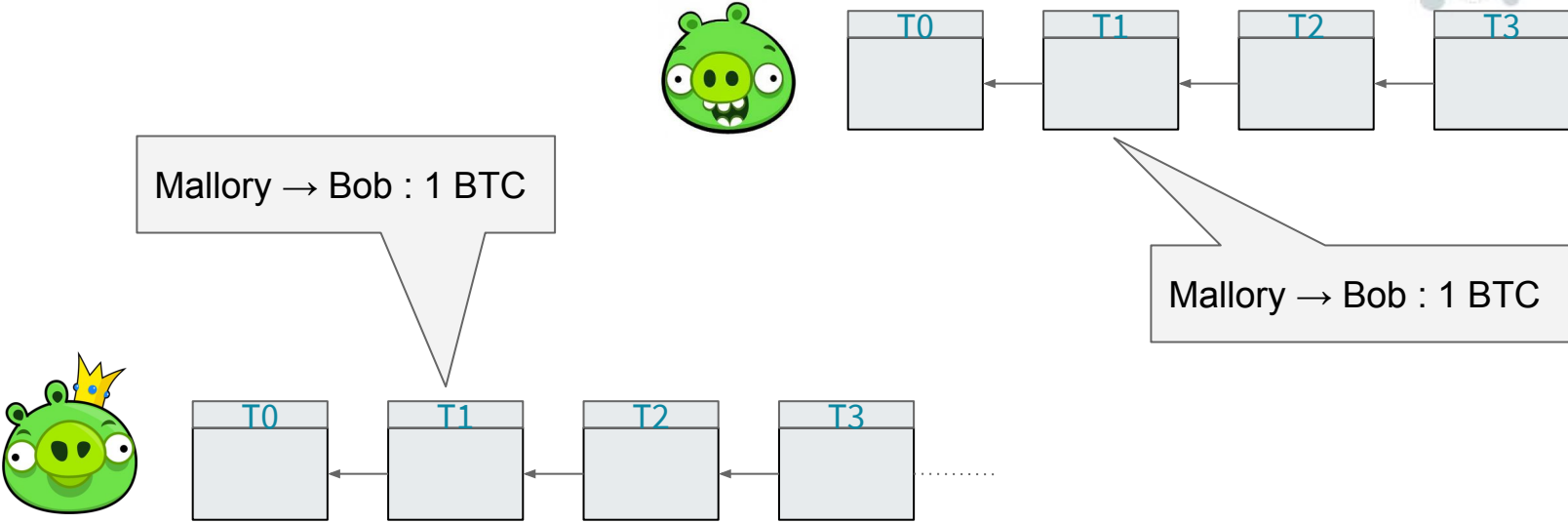
# Blockchain consistency



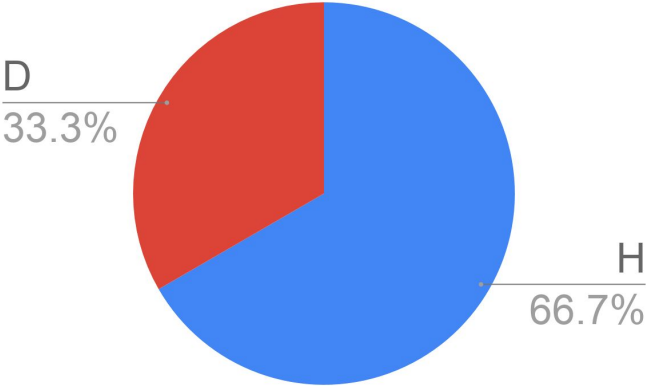
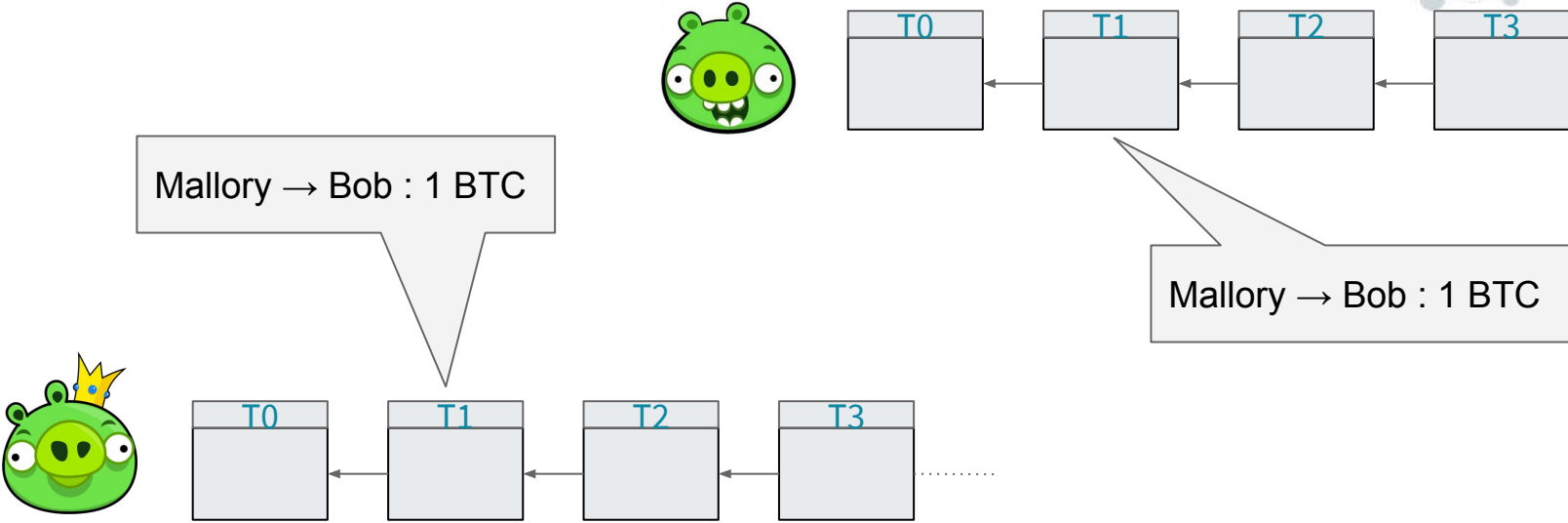
# Blockchain consistency



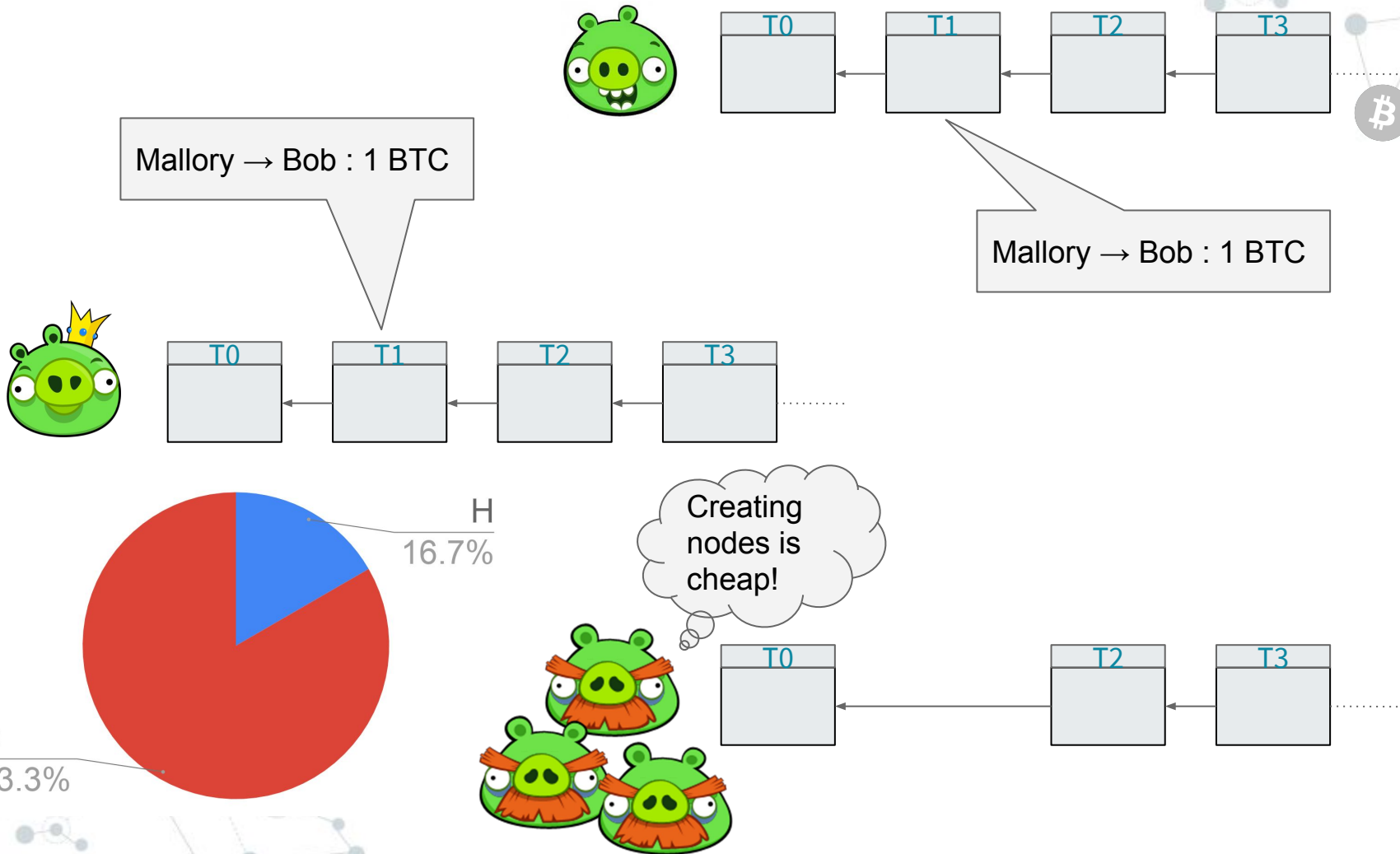
# Blockchain consistency



# Consistency "by majority"

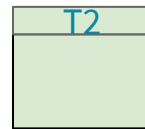
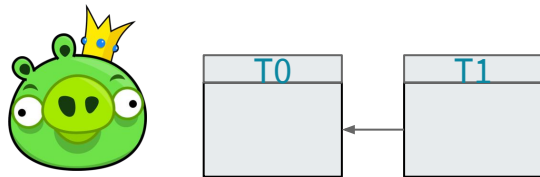
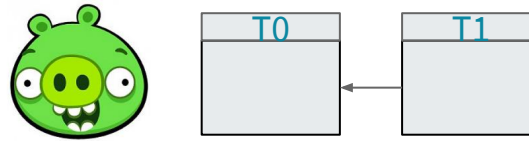


# Consistency "by majority": Sybil attacks!

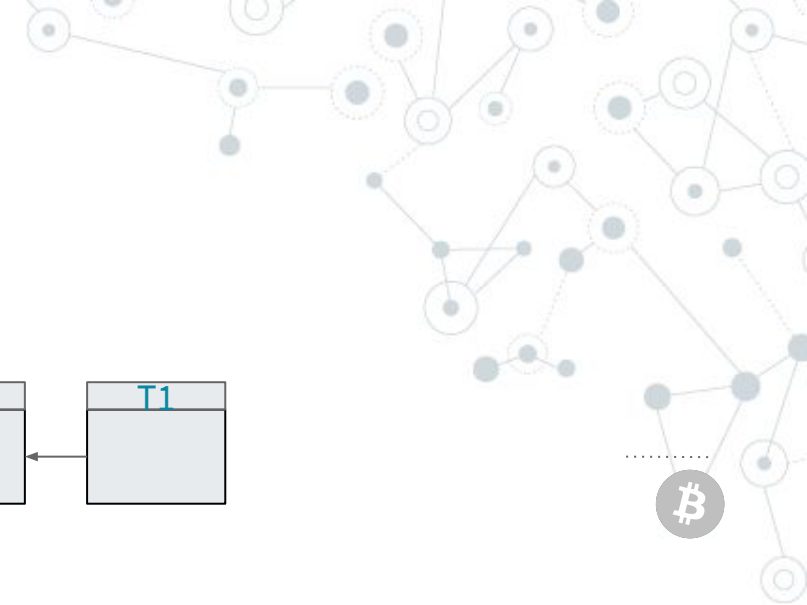
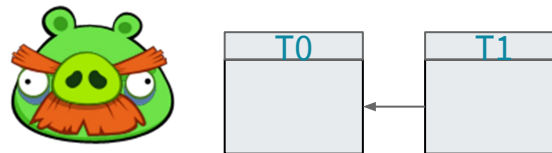




# Consistency by “proof-of-work”



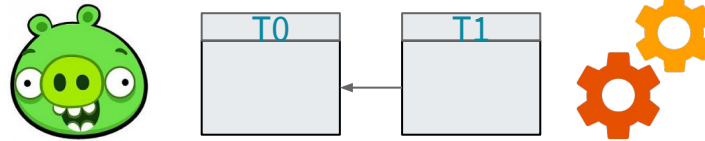
Alice → Bob : 1 BTC



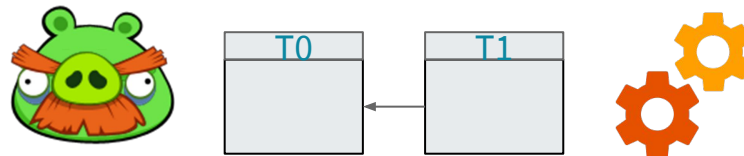
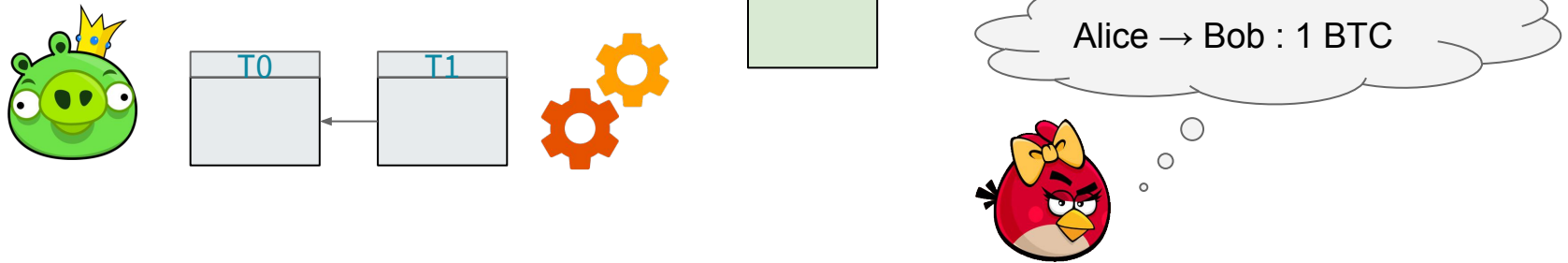
.....

# Consistency by “proof-of-work”

Miners can freely set  $r$  bits within  $T_2$ .



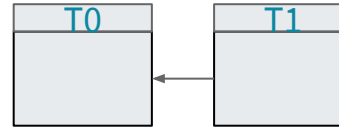
The protocol fixes a constant  $c$  (difficulty).



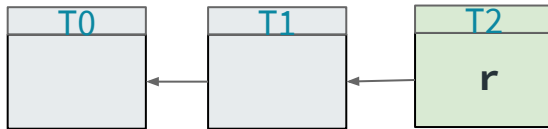
# Consistency by “proof-of-work”

## Proof-of-work:

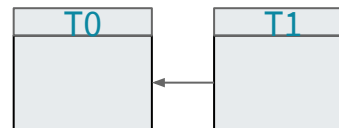
T2 can be appended only if  $\text{hash}(T2) < c$



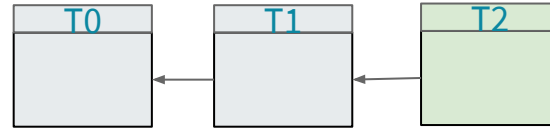
~10' to find suitable r



Alice → Bob : 1 BTC



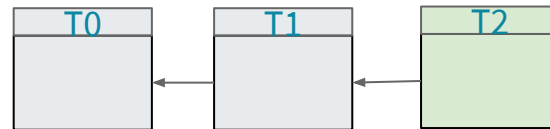
# Consistency by "proof-of-work"



Alice → Bob : 1 BTC



The miner who solves the puzzle wins some coins (and the transaction fees)



## Beyond currency transfers: embedding metadata

Transactions can store 80 bytes of arbitrary data:

T
in: ...
wit: ...
out(x): <b>OP_RETURN</b> <80 bytes>
val: 0 BTC

T can be appended to the blockchain, but its output can *not* be spent by **any** transaction

## Beyond currency transfers: embedding metadata

June 20, 2018 12:20 AM

Dear blockchain, please let me quit smoking.



June 1, 2018 3:28 PM

ciao mamma



## Beyond currency transfers: embedding metadata

Select a document and have it certified in the Bitcoin blockchain [What?](#)

Click here or drag and drop your document in the box.

The file will NOT be uploaded. The cryptographic proof is calculated client-side.

### Congratulations!

This document's digest was successfully embedded in the Bitcoin blockchain. It is now permanently certified and proven to exist since the transaction was confirmed.

Transaction [b37d3533e55c4d54075dc7e71d698c8196fd55b77a94d0a06e4d515606cb53b1](#)

# Beyond currency transfers: embedding metadata



## TRANSACTION INFORMATION

738fc5d16b533ccedd775ad7395f14...

Sent **0.00006**

Asset Sent **3.86031**



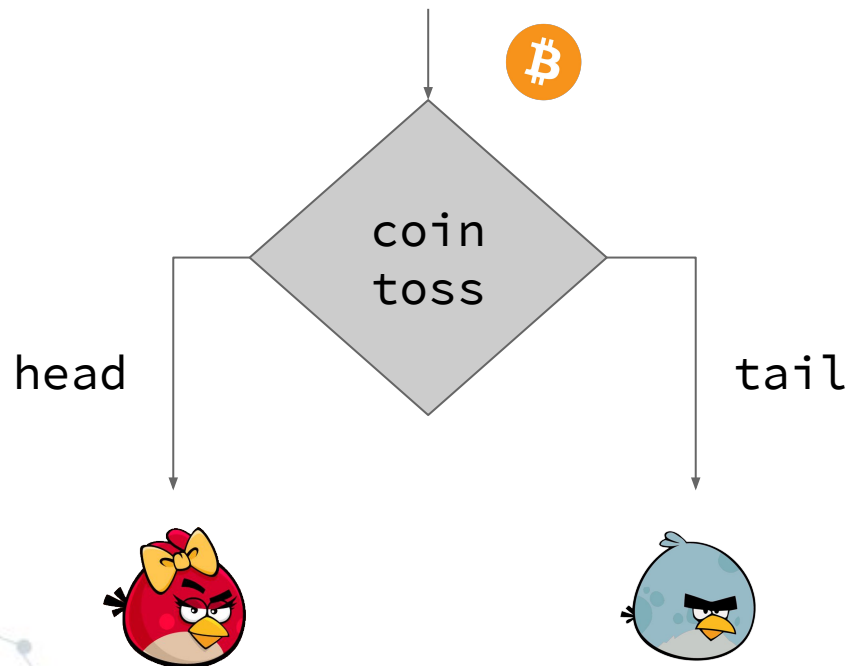
Inputs	Amount	Asset	Outputs	Amount	Asset
< <u>1HwgN6Kkwj5CUq6...</u>	0.00003	BTC	<u>18B6kkLRAMBjdTN...</u>	0.00003	BTC
	3.86031	 <u>La42SrD4H9LuAznWS5p3...</u>		3.86031	 <u>La42SrD4H9LuAznWS5p3...</u>
< <u>1HwgN6Kkwj5CUq6...</u>	0.00043	BTC	N/A	0	BTC
			<u>1HwgN6Kkwj5CUq6...</u>	0.00003	BTC

Raw HEX



## Beyond currency transfers: smart contracts

Bitcoin contracts are cryptographic protocols to transfer BTC. The consensus protocol of the blockchain guarantees their secure execution.



## Beyond currency transfers: smart contracts

- Oracles (feeds of external data to the blockchain)
- Escrow and arbitration
- Crowdfunding
- Micropayments channels (“Lighting network”)
- Lotteries & other gambling games (Poker, ...)
- ...

More complex contracts are possible using off-chain cryptographic protocols (ZK proofs)

A decorative background featuring a network diagram of nodes and connections. The nodes are represented by circles of varying sizes and colors (grey, blue, and white with a blue outline). Some nodes contain a Bitcoin symbol (₿). The connections are thin grey lines. The network is more dense on the left and right sides, with the central area being mostly white space containing the text.

# Bitcoin problems

a non-exhaustive list

## Issue #1: too much anonymity

Carl



I want to transfer 1 BTC to myself, **anonymously**

Masqueraded Carl



T
in: ...
wit: ...
out(x): $\text{ver}_{\text{Carl}}(x)$
val: 1 BTC

# Issue #1: too much anonymity

Carl

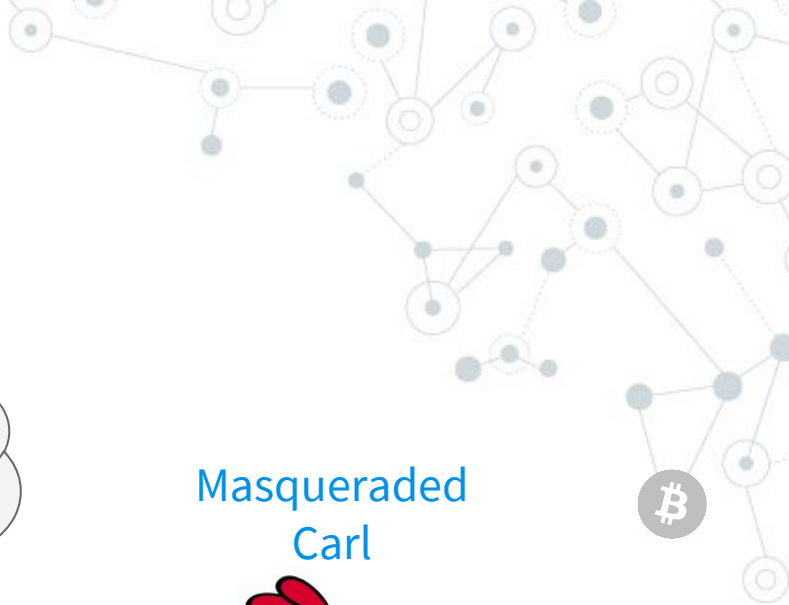


Who am I?  
... just a key pair  
 $K = (K_s, K_p)$

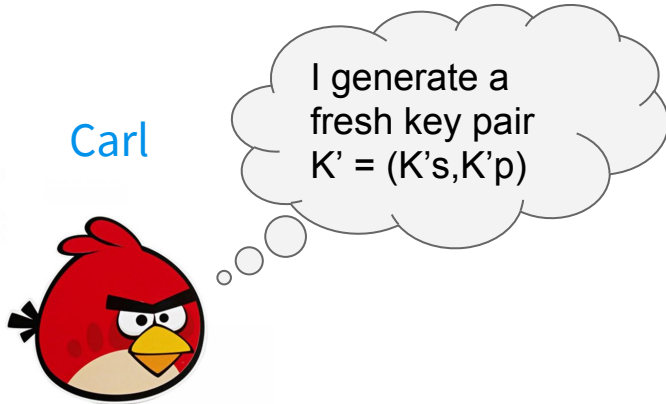
Masqueraded  
Carl



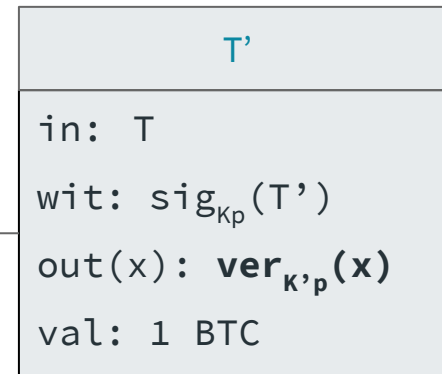
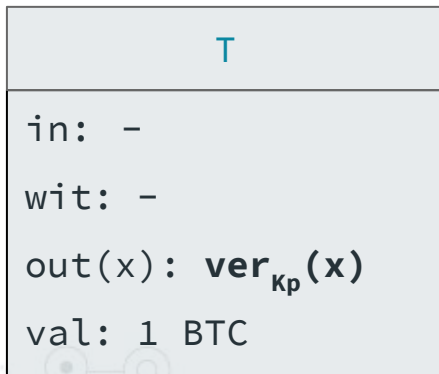
T
in: ...
wit: ...
out(x): $\text{ver}_{K_p}(x)$
val: 1 BTC



# Issue #1: too much anonymity



Masqueraded Carl



**Bitcoin address** = pseudonym = hash of public key

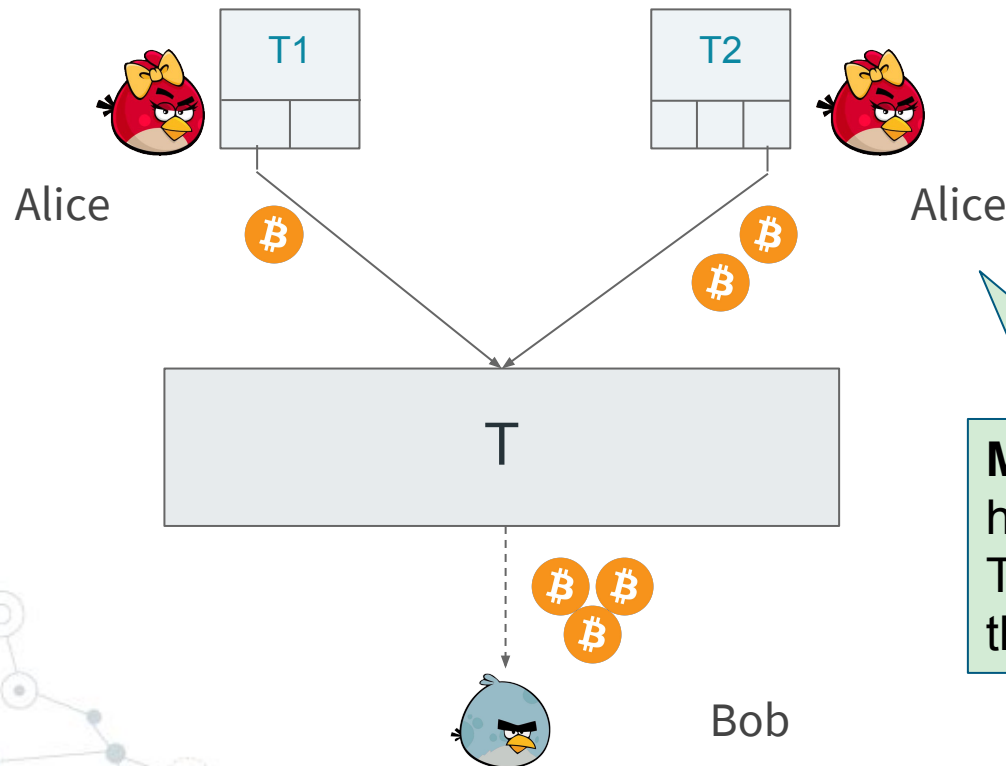
## Issue #1: too much anonymity



“Killer” criminal applications: Ponzi schemes, money laundering, crypto-lockers, ...

Issue #2: not enough anonymity (criminals' point of view)

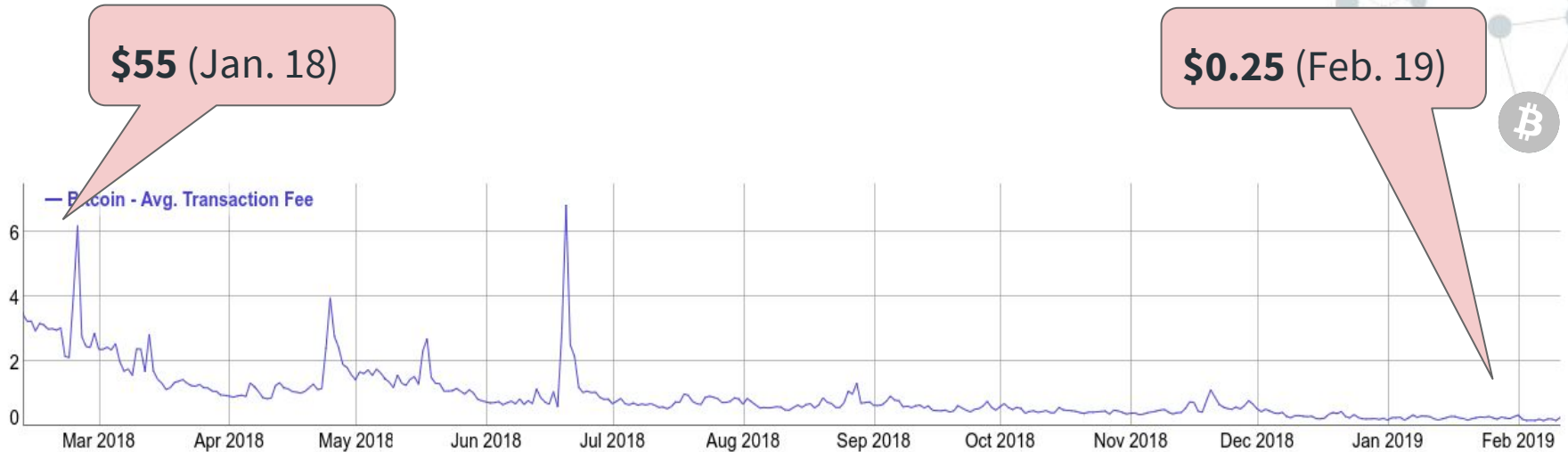
**Address clustering** techniques try to group addresses controlled by the same entity.



**Multi-input**  
heuristics: inputs of  
T are controlled by  
the same entity



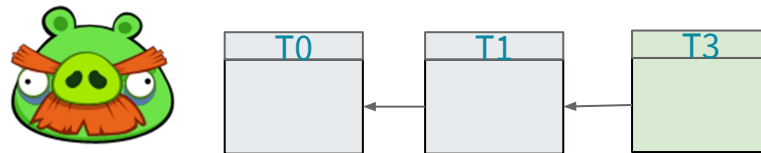
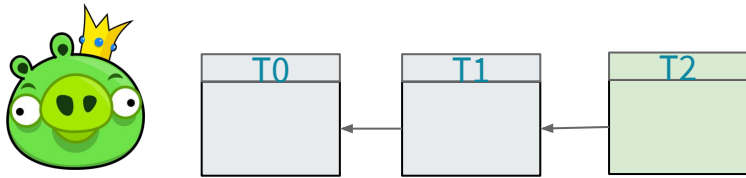
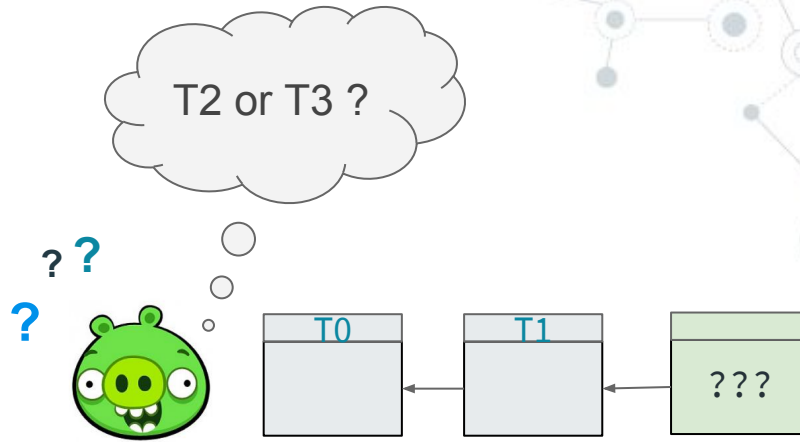
## Issue #3: unpredictable transaction fees



- ◎ Fees depend on Bitcoin market
- ◎ dApps built upon Bitcoin must take fees (and their variability) into account

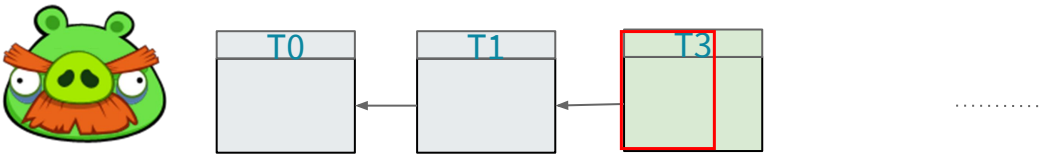
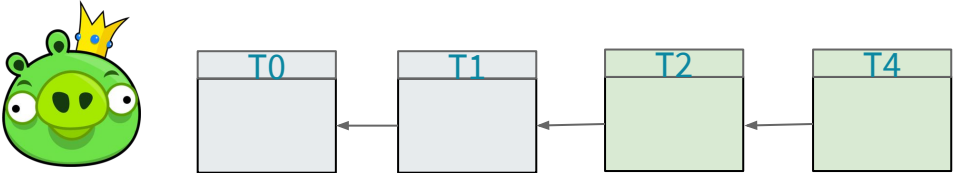
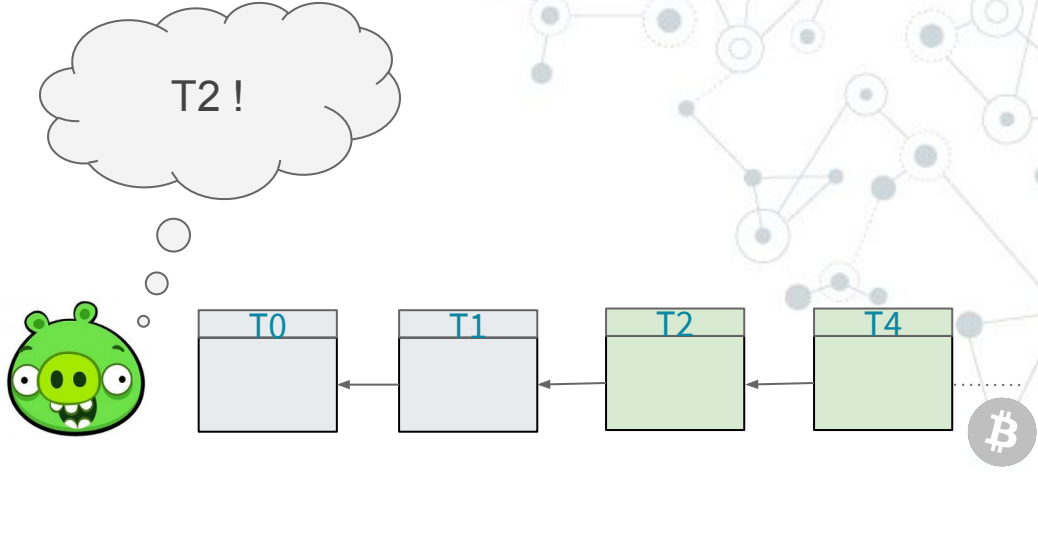
# Issue #4: forks

Two miners may solve the puzzle almost simultaneously



# Issue #4: forks

The **longest** chain always wins



## Issue #5: low throughput & high latency

A new block is added to the blockchain every **10 minutes**

The price of removing a block **B** from the blockchain grows exponentially in the number of blocks appended after **B**

Usually, a transaction is considered **confirmed** if it has been published in a block with at least **5 subsequent blocks**

**latency = ~ 60 m**

Each block contains **~2000** transactions.

**throughput = ~ 3 tx/s**

(VISA: ~2700 tx/s)

## Issue #6: speculation

Main use case of Bitcoin: **speculative investment**

This has several drawbacks:

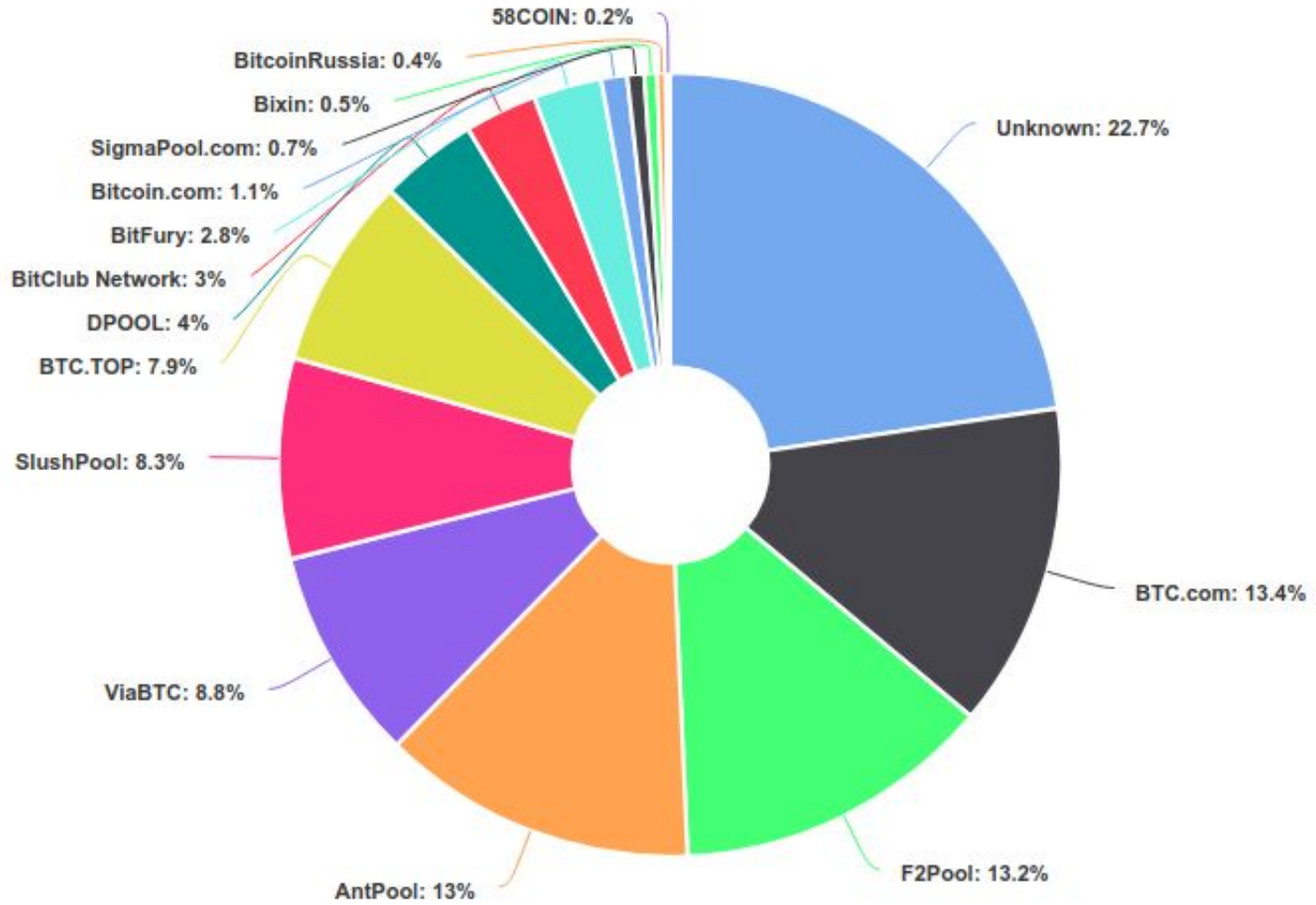
1. Fees may grow with speculation
2. Governance issues  $\Rightarrow$  resilience to innovation
3. Bitcoin may cease to exist when the bubble pops
4. ...



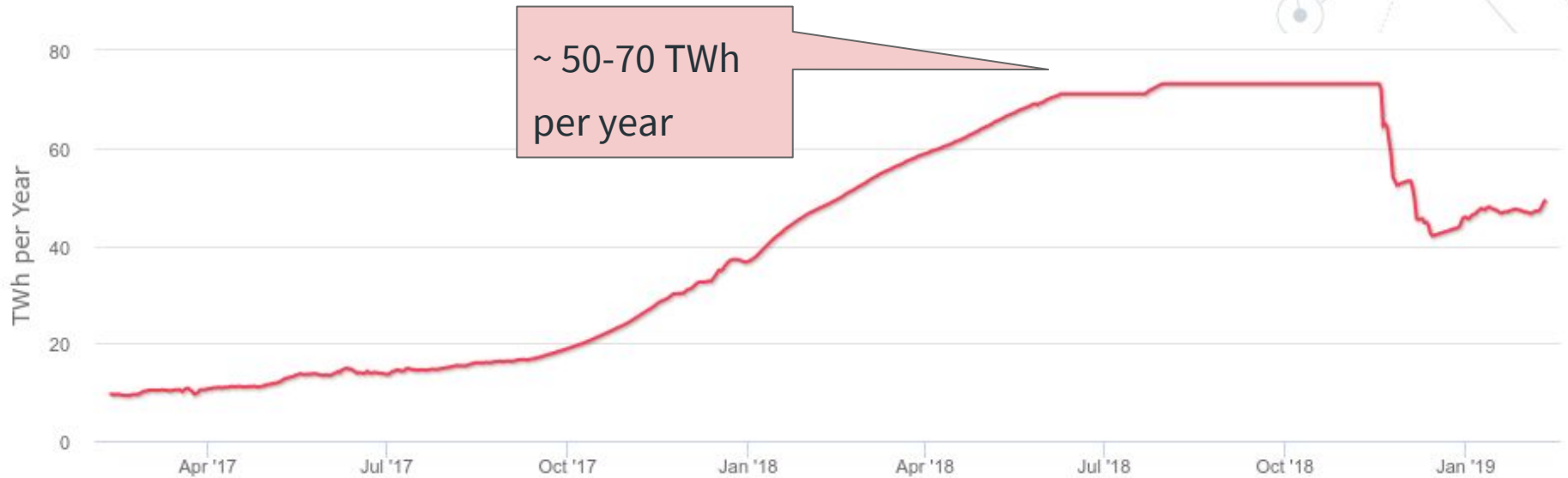
## Issue #7: limited expressiveness of smart contracts

1. Bitcoin allows for simple (yet useful) smart contracts
2. Successful Ethereum contracts tend to be complex:
  - a. Decentralized exchanges (IDEX, ForkDelta, Bancor, ...)
  - b. Pseudo-Ponzi games (CryptoKitties, PowH3D, Fomo3D, ...)
  - c. ...
3. A few extensions of the scripting language would be enough to enhance expressiveness of contracts
4. Little hope that proposals for extensions will be accepted!

## Issue #8: centralization (!)



## Issue #9: energy consumption / pollution



1. Singapore: ~48 TWh per year (~ BTC)
2. Italy: ~300 TWh per year
3. World: ~17K TWh per year (BTC ~ 0.3% world)



## Issue #10: immutability (!)

Since Bitcoin transactions can embed arbitrary data, they can also contain **illegal** data (Matzutt et al., FC 18):

1. Child pornography
2. Blasphemous material
3. Non GDPR-compliant data
4. ...

It is **impossible** to remove illegal data once they are on the blockchain!



# Post-Bitcoin blockchains

A decorative background featuring a network of interconnected nodes and lines, with several nodes containing a Bitcoin symbol (₿).

## Different applications require different blockchains

1. Who can write? (anyone, predefined set of nodes, ...)
2. Who can read? (anyone, restricted set of nodes)
3. Consensus (PoW, PoS, BFT, ...)
4. Latency / throughput
5. Privacy
6. Expressiveness of the scripting language
7. Transaction fees

## Bitcoin

1. Who can write? anyone  $\Rightarrow$  **permissionless**
2. Who can read? anyone  $\Rightarrow$  **public**
3. Consensus: **PoW**
4. Latency / throughput : **high / low**
5. Privacy: **pseudonymity**
6. Expressiveness of scripting: **low**
7. Transaction fees: **high**



## Zcash

1. Who can write?
2. Who can read?
3. Consensus
4. Latency / throughput
5. Privacy: **anonymity**
6. Expressiveness of scripting: low
7. Transaction fees: **high**

same as Bitcoin  
(piggy-back on the  
Bitcoin blockchain)



## Ethereum

1. Who can write? anyone  $\Rightarrow$  **permissionless**
2. Who can read? anyone  $\Rightarrow$  **public**
3. Consensus: **PoW** (switching to **Proof-of-Stake?**)
4. Latency / throughput: **high** / **low** (but better than BTC)
5. Privacy: pseudonymity
6. Expressiveness of scripting:  $\sim$  **Turing-complete**
7. Transaction fees: **high**



## HyperLedger Fabric

1. Who can write? predefined nodes  $\Rightarrow$  **permissioned**
2. Who can read? custom  $\Rightarrow$  **public / private**
3. Consensus: custom (**PBFT**)
4. Latency / throughput: **low / high**
5. Privacy: **none** (all writers are known)
6. Expressiveness of scripting: **Turing-complete**
7. Transaction fees: **0 (no cryptocurrency)**



## AlgoRand

1. Who can write? anyone  $\Rightarrow$  **permissionless**
2. Who can read? anyone  $\Rightarrow$  **public**
3. Consensus: **Proof-of-Stake (NO FORKS)**
4. Latency / throughput: **low / high**
5. Privacy: pseudonymity
6. Expressiveness of scripting: **??**
7. Transaction fees: **0??**





A decorative background featuring a network of interconnected nodes and lines, with several nodes containing a Bitcoin symbol (₿).

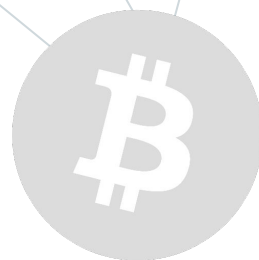
Do you really need a blockchain?

What can be done with a blockchain that **cannot** be done with a centralized database?

**NOTHING**

it's all a matter of **TRUST**

blockchains = trust the **crowd**, not the single



# Thank you

Blockchain Summer School @ Pula (CA), 10-14 June 2019  
(max 30 students, free, funded by Sardegna Ricerche)

Gruppo di lavoro CINI su DLT: <http://dltgroup.dmi.unipg.it/>