UNIVERSITÀ
DEGLI STUDI
DI UDINE

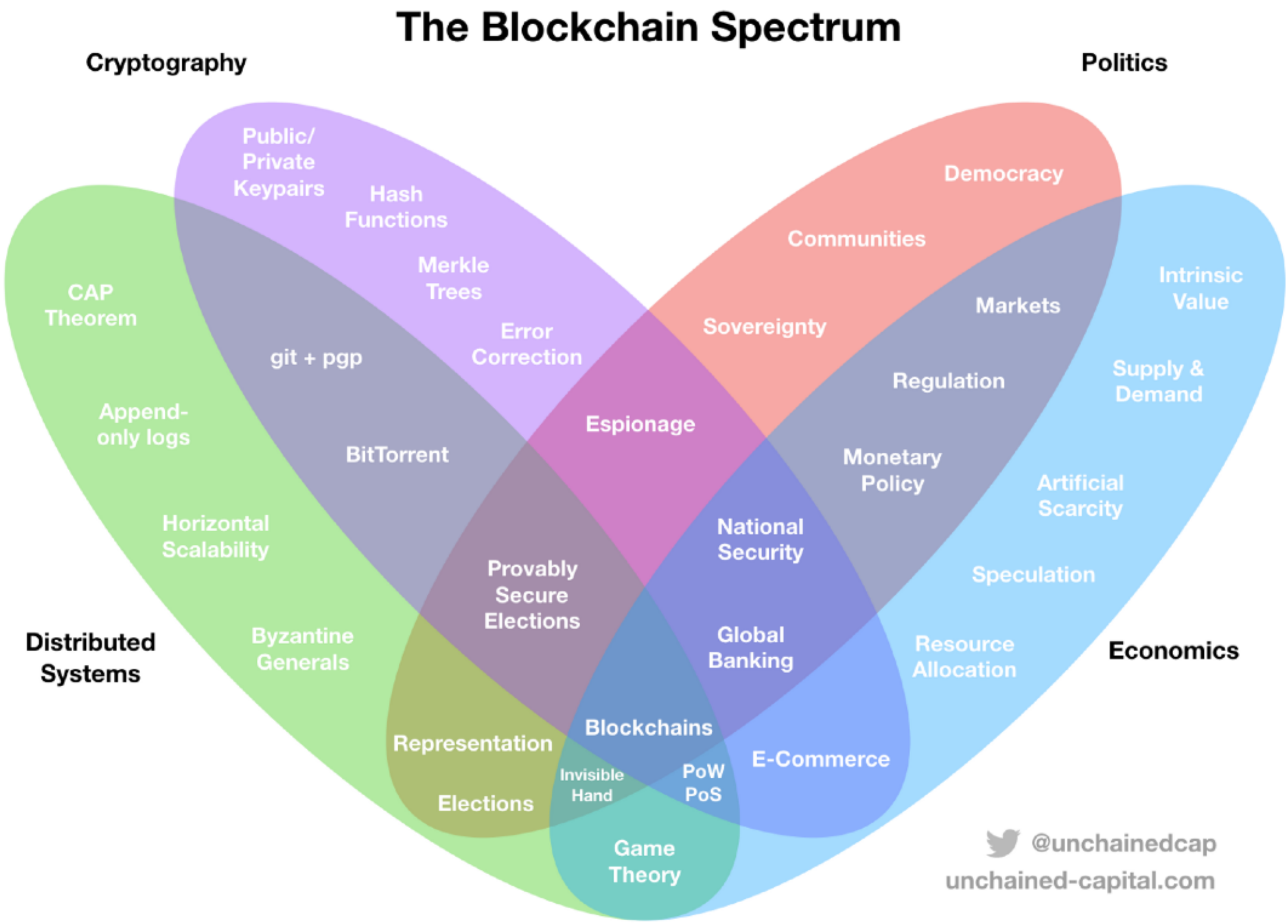
Dai generali bizantini alla blockchain: come fidarsi quando non c’è da fidarsi

Prof. Marino Miculan
Models and Applications of Distributed Systems Lab
DMIF, Università di Udine

Campus Informatica 2022 — 29 agosto 2022

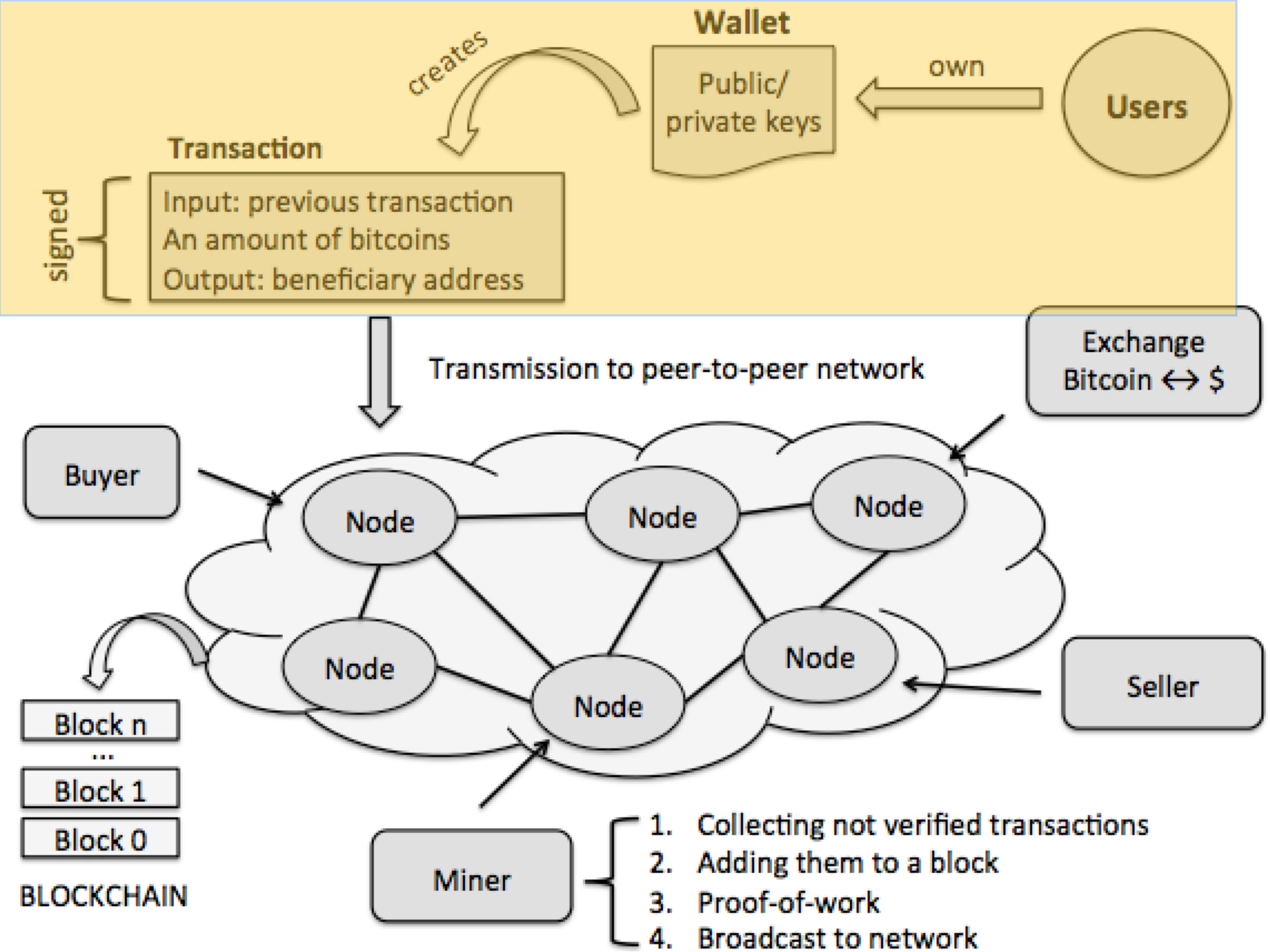
How to explain Bitcoin and Blockchain?

- Explaining Bitcoin in short form is no easy task.
- Involves
  - Cryptography
  - Computer networking and data transmission
  - Game theory
  - Economic and monetary theory
- But foremost, a **cultural paradigm shift: It could replace any processing central authority with decentralized peer-to-peer cryptographically secure equivalent**



What is Blockchain, In One Slide

- **Transactions**: Transfers of bitcoin from **input addresses** to **output addresses**
- **Blocks**: Timestamped collection of transactions.
- **Miner**: Agent which validates transactions and puts them into blocks
- **Blockchain**: The entire series of blocks 'chained' together
    - Miners compete to add blocks, the "winner" is compensated with bitcoins



Transactions: transfers of what?

- In fiat money, transaction is exchange of physical object (coins, bills, …).
    - Ownership is on owns the physical object

- But bitcoins exist only virtually. No physical object to exchange
- Instead: **a transaction is a common and shared agreement on change of ownership**
    - Bitcoins are not “moved”. Only their ownerships change.
    - Ownership can be split among different participants, or merged

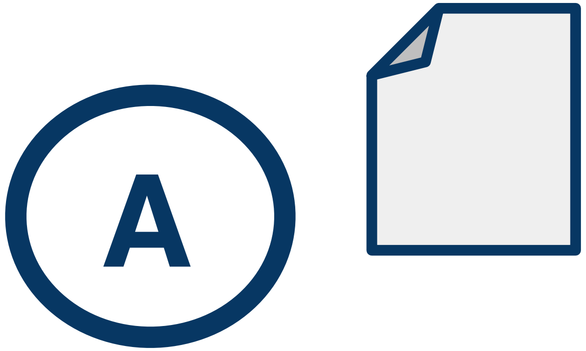
An analogy: Rai stones (Yap islands, 15th cent.)

- Rai stones are carved and placed somewhere, then never moved
- The monetary system relies on an **oral, shared history of ownership.**
    - Buying an item with a rai stone simply involves **agreeing that the ownership has changed.**
    - The transaction is recorded in the oral history, shared with all the village. *Sharing is caring!*
    - No physical movement of the stone is required
    - In fact, *no physical access to the stone is needed*: some stones went "lost" into the sea, but they still have been used in transaction anyway.
- See https://youtu.be/J-ab9was1p0

The Ledger

- We have seen what are transactions, and how they work
- Now the problem is: where we store these transactions?
- We need a **permanent ledger**
- Basic requirements:
  - **Consistent**: what it says is the truth, so it must be correct
  - **Immutable**: once a transaction has been validated, it cannot be retracted
  - **Available**: high availability, 24/7, and impossible to delete/lose
  - **Scalable**: on a global scale, with billions of users and transactions
  - It can be public, if transactions achieve privacy by other means (like the pseudonyms we have seen)

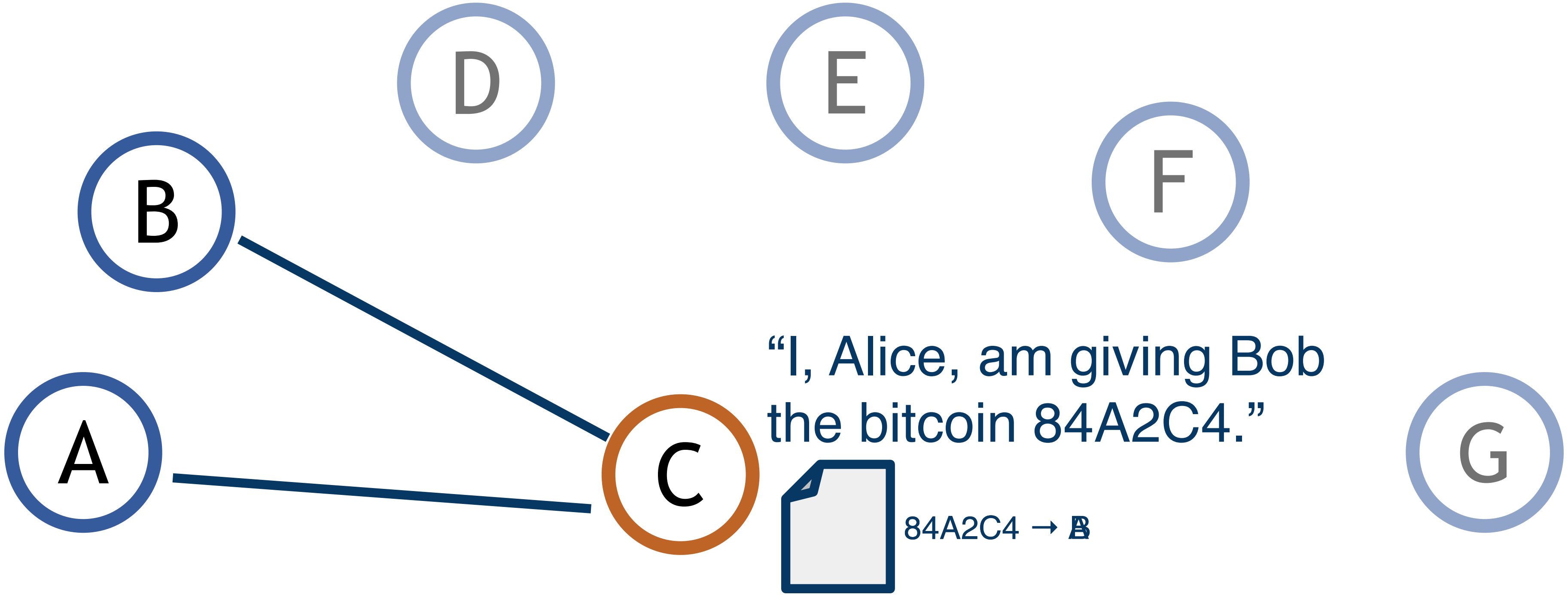
- Let’s see an example of how it works.
  In a naive version of the protocol, Alice writes and signs a message describing her transaction



“I, Alice, am giving Bob
the bitcoin 84A2C4.”

Avoiding double spending: centralised ledger

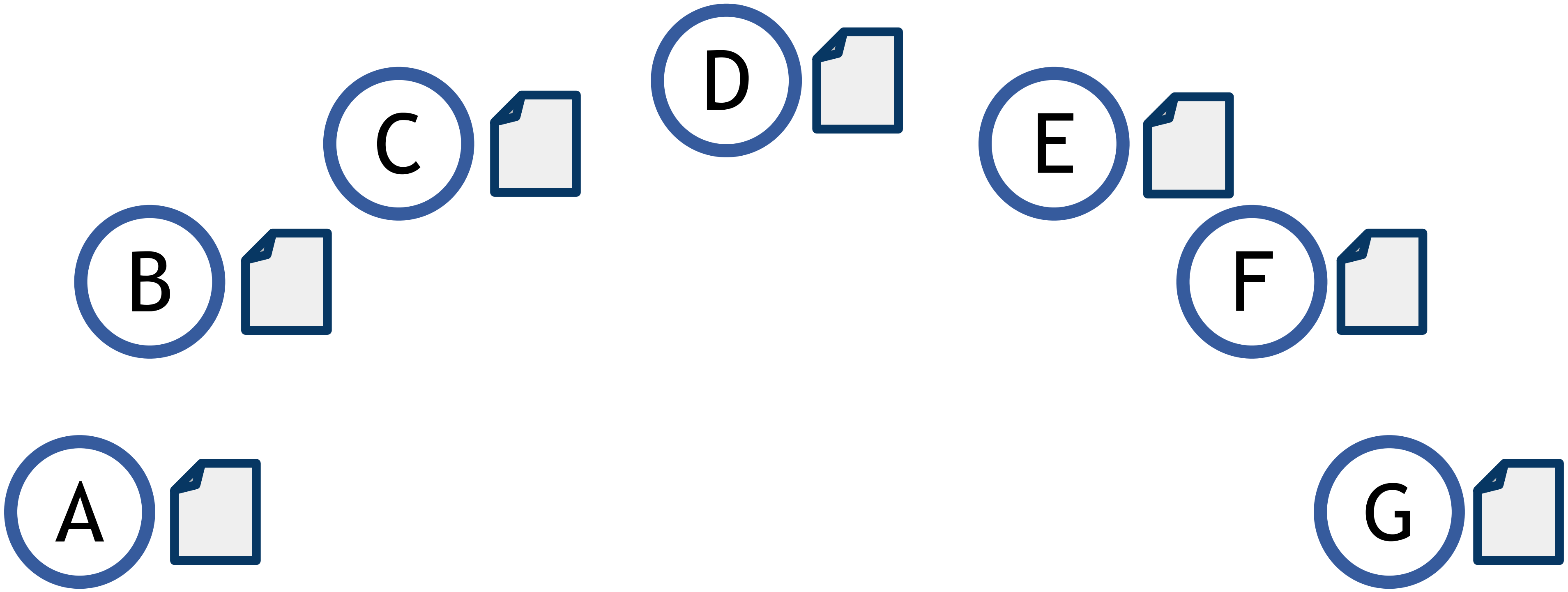
- In a centralized solution, a central node (the bank, or some other agency) manages transactions and balances
- Solves the issue of duplication, but we must trust the bank!



“I, Alice, am giving Bob the bitcoin 84A2C4.”

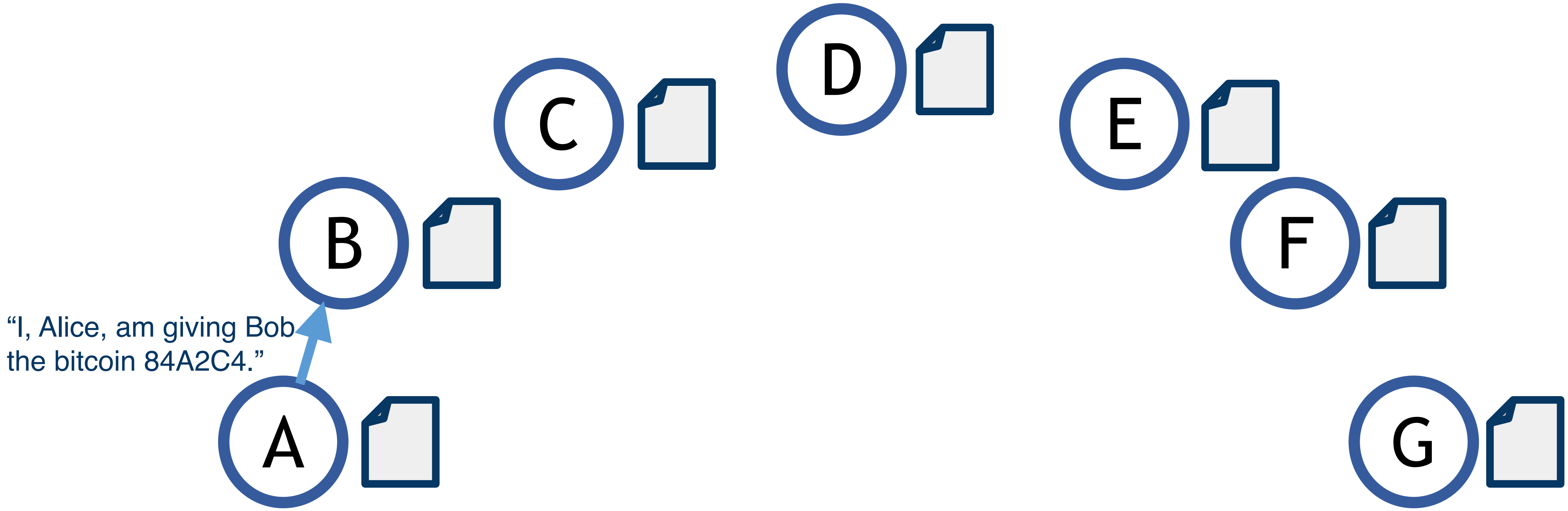
Avoiding double spending: decentralised ledger

- Decentralization: Making everyone the bank
- Every node has a complete copy of all transactions



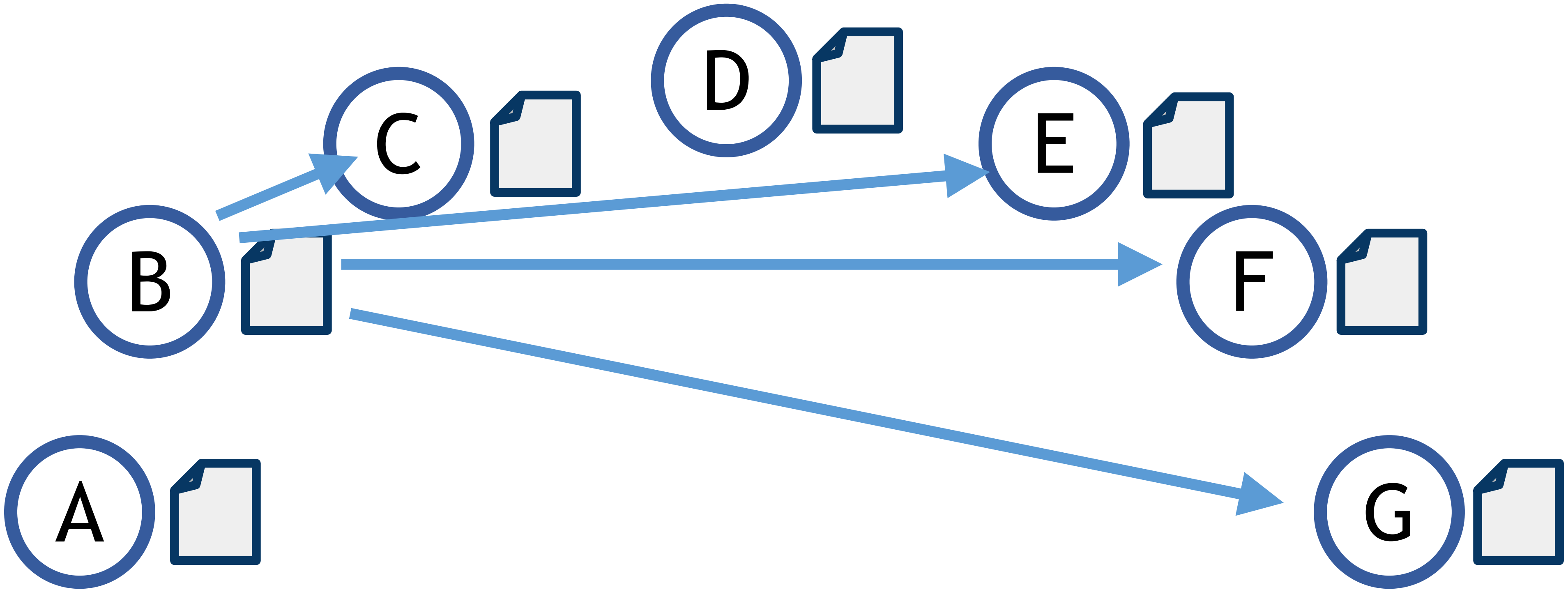
Avoiding double spending: decentralised ledger

- Alice sends her transaction to Bob
- He can check his copy of the blockchain to be sure that the bitcoin actually belonged to Alice



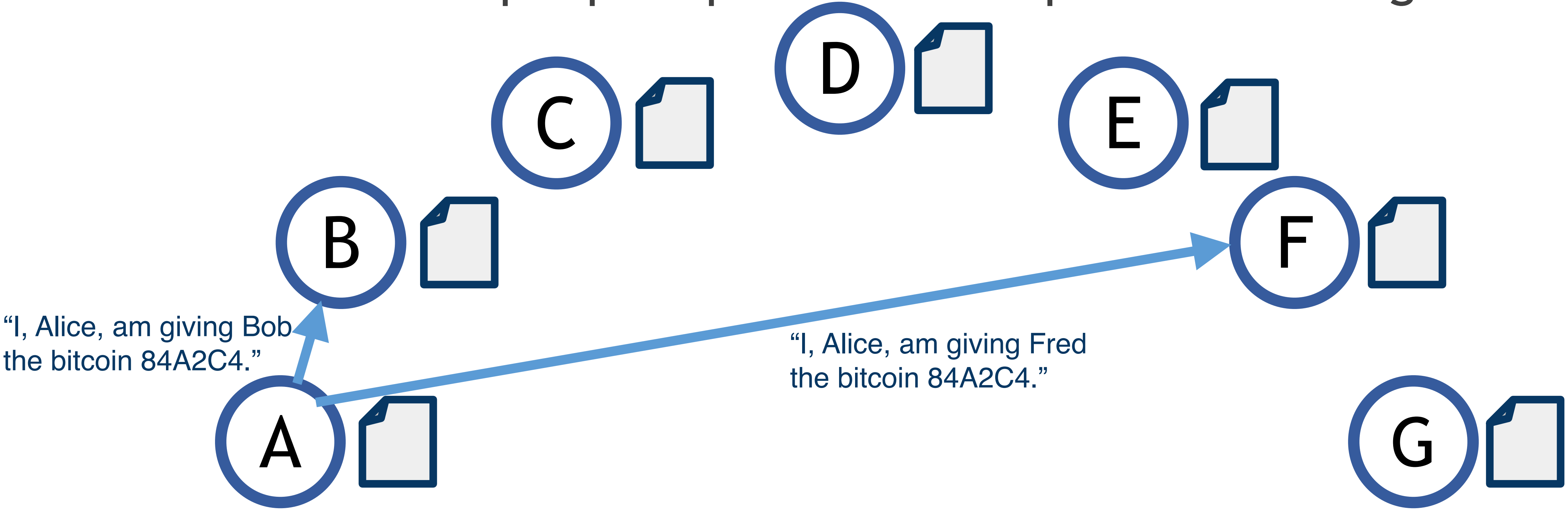
Avoiding double spending: decentralised ledger

- If that works, Bob announces the transaction to the world
- Every node checks the transaction with its copy of the ledger, and updates it



Avoiding double spending: decentralised ledger

- But what if Alice double spends the same bitcoin on Bob and Fred?
- Both will find that the bitcoin belonged to Alice, accept the transaction, and announce it to the world.
- How should other people update their copies of the ledger?

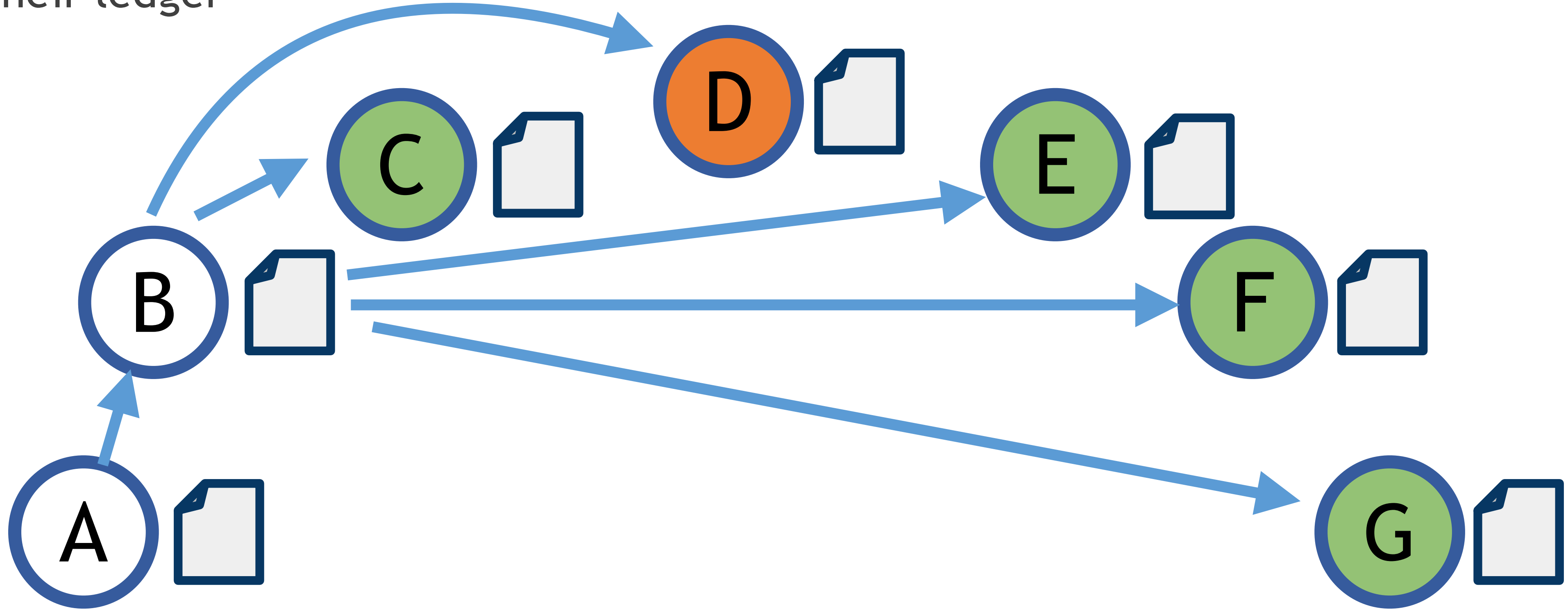


"I, Alice, am giving Bob the bitcoin 84A2C4."

"I, Alice, am giving Fred the bitcoin 84A2C4."

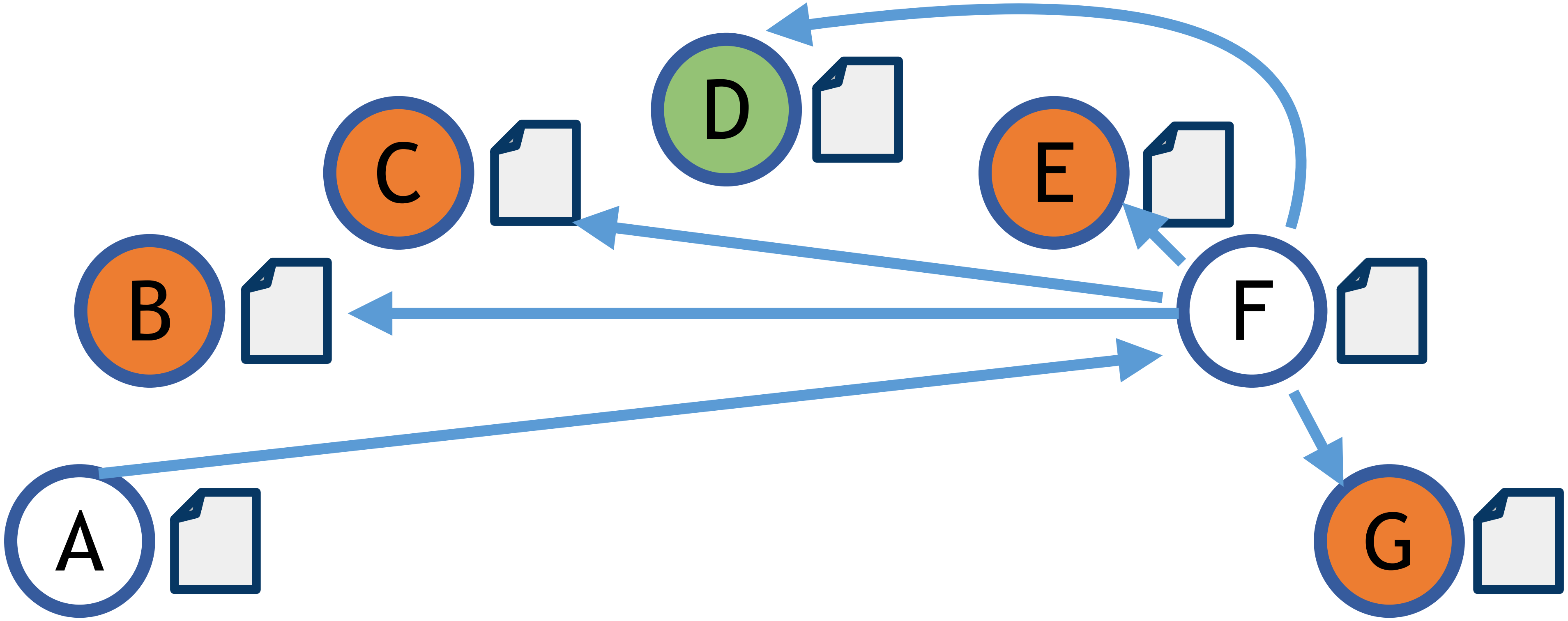
Avoiding double spending: decentralised ledger

- Solve this problem by letting everyone verify transactions
- When Alice sends her transaction to Bob, Bob broadcast the possible transaction to the entire network of bitcoin users, and ask them to verify it wrt their copy of the ledger.
- If enough users verify the transaction, Bob can accept the bitcoin, and everyone will update their ledger



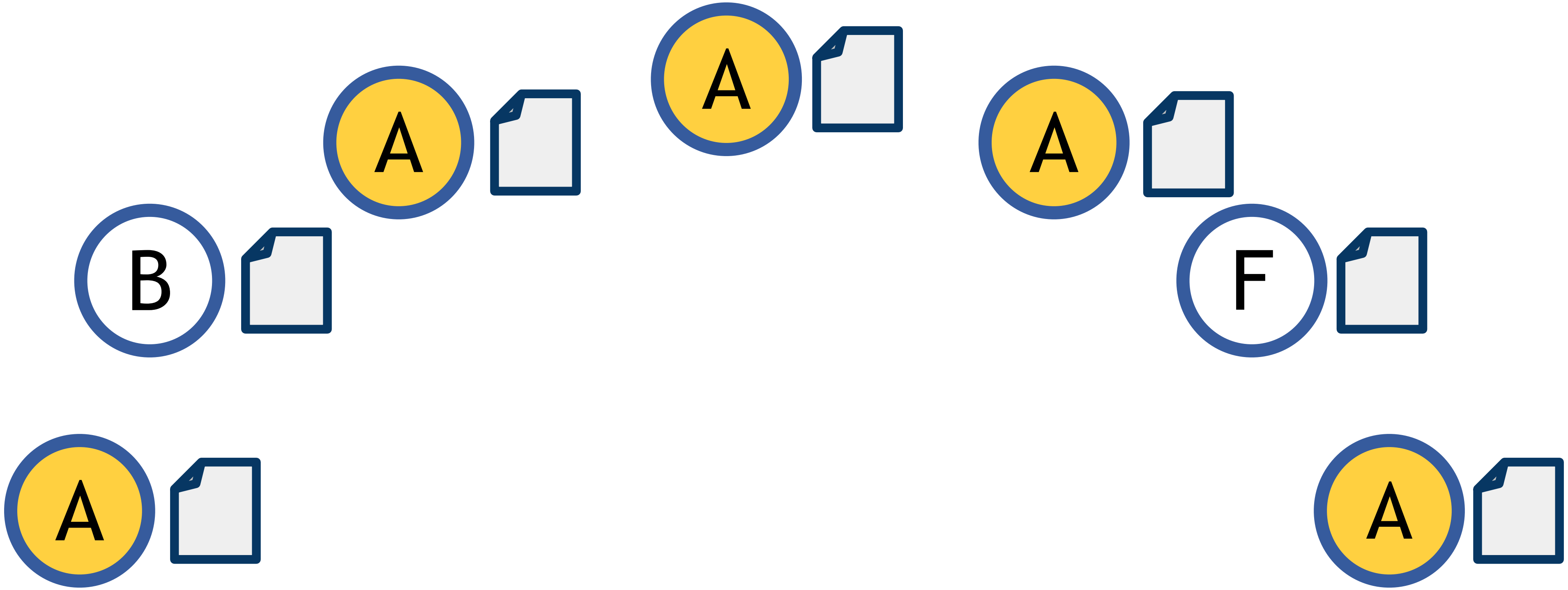
Avoiding double spending: decentralised ledger

- If Alice tries to double spend on Bob and Fred, the first transaction is validated but not the second, so Fred is not fooled



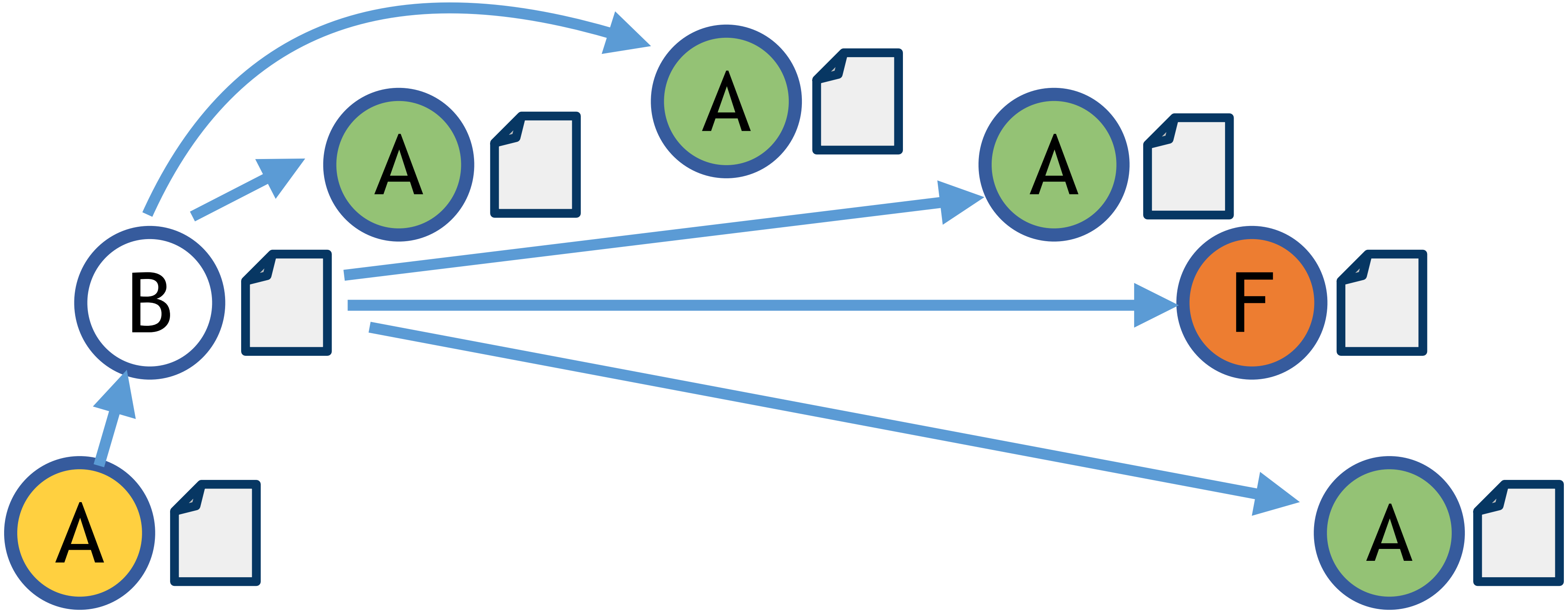
Decentralised ledger is not enough…

- “**Sybil attack**”: Alice can set up a majority number of separate identities that overwhelm the bitcoin network
- These fake users can act in whatsoever way Alice decides



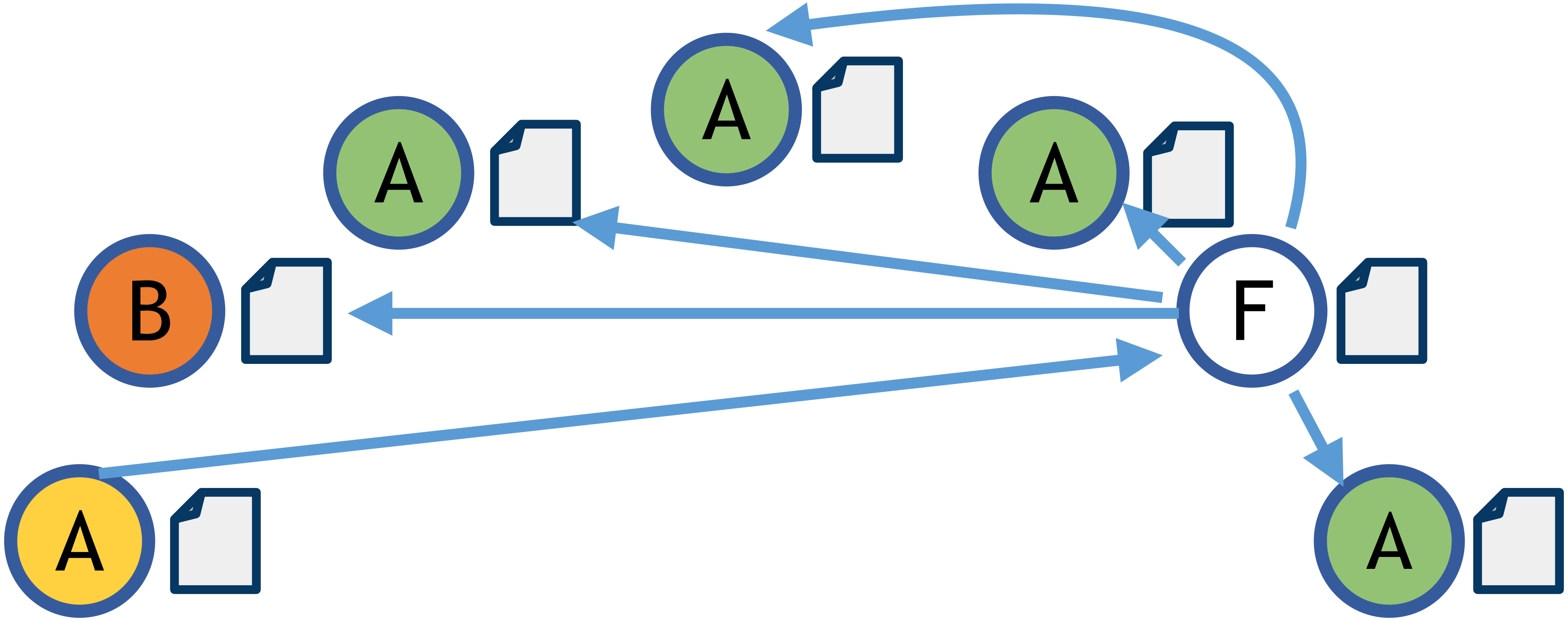
… if many nodes are byzantine

- Alice double spends on Bob and Fred
- The fake identities confirm both spendings, so B and F are fooled!
- A's copies are **byzantine**



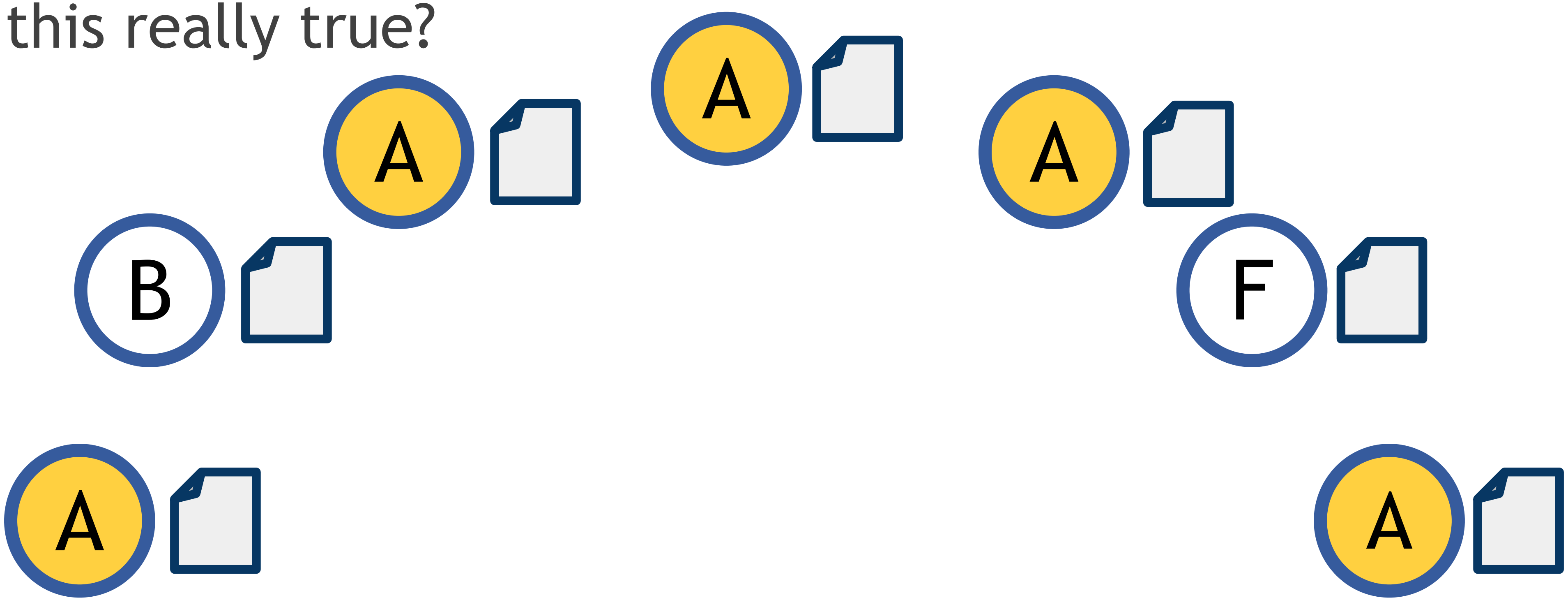
… if many nodes are byzantine

- Alice double spends on Bob and Fred
- The fake identities confirm both spendings, so B and F are fooled!
- A’s copies are **byzantine**



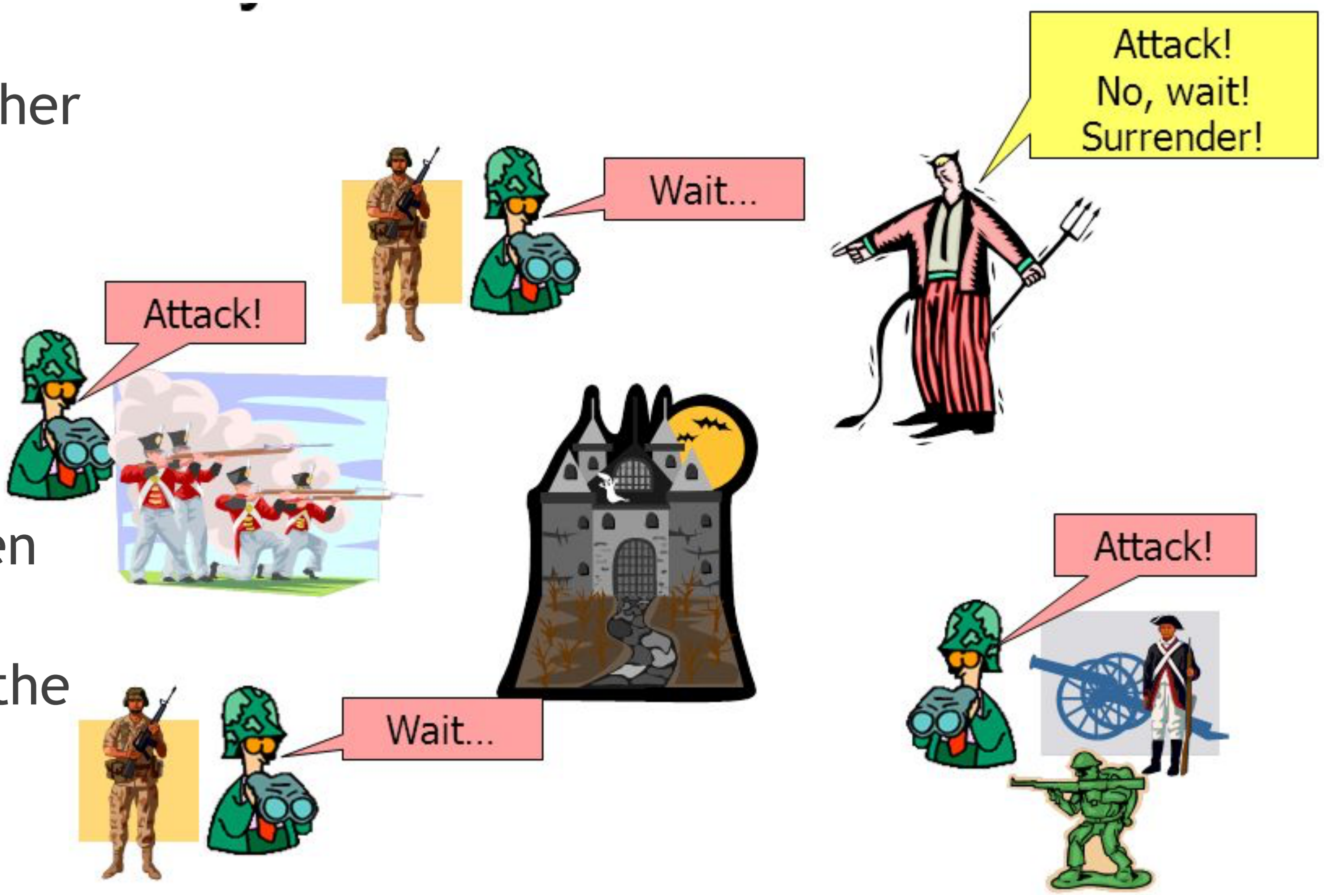
Would simple majority work?

- In a “democratic” system, we would decide to assign the bitcoin to either B or F by asking all nodes and looking at the majority
- In this way the dishonest nodes should be 51% of the whole, in order to carry out the Sybil attack
- But is this really true?



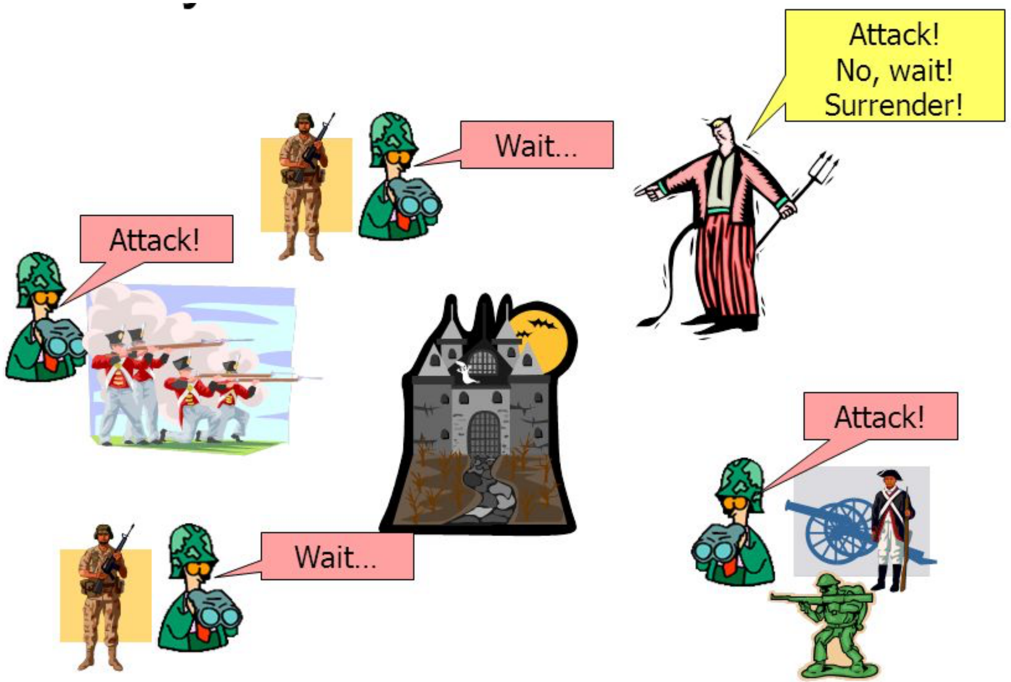
The Byzantine Generals Problem [Lamport, 1982]

- A general and *n-1* lieutenants have to decide to either attack or not a sieged fortress
- General and lieutenants can talk to each other
- Attack is successful if the majority of lieutenants attack together.
- **But *some* participants are traitors!**
  - A traitor general may give different orders to different lieutenants
  - A traitor lieutenant may not attack when he should
  - Loyal participants don't know who are the traitors!! Trust nobody!
- **How to reach agreement among the loyal participants?**

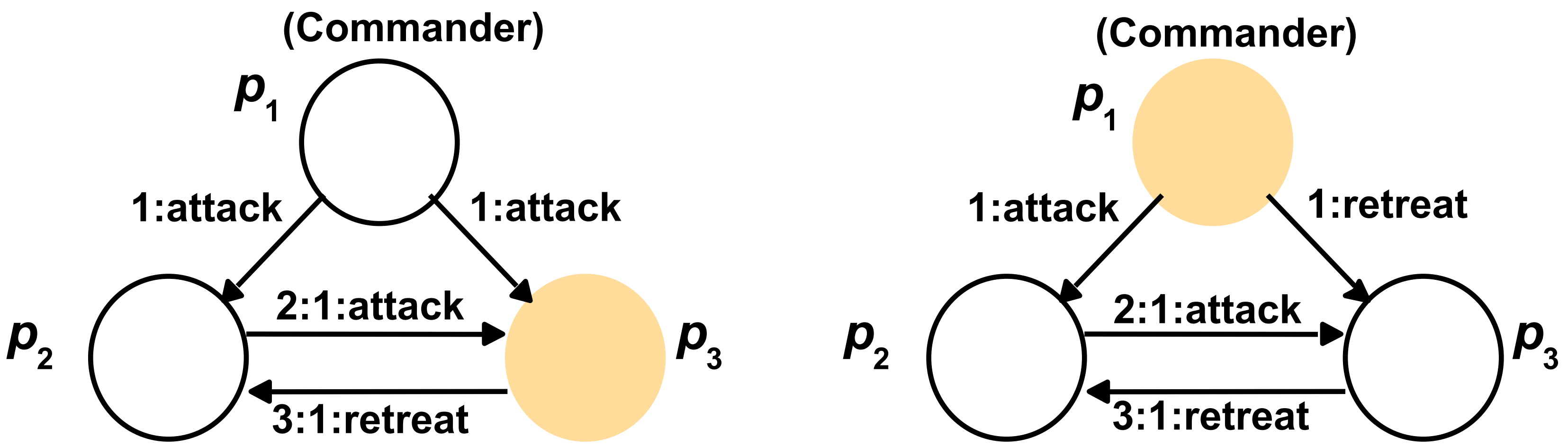


The Byzantine Generals Problem [Lamport, 1982]

- Theorem: **Agreement of loyal parties can be achieved only if traitors are strictly less than 1/3 of the whole**
    - E.g., if n=3 (1 general + 2 lieutenants), one traitor is enough to confuse the other two
    - if n=12 (1 g.+11 l.), 4 traitors are enough to confuse the remaining 8
- **Simple majority of honest nodes is not enough!**
- This result is important in designing fault tolerant systems



Three byzantine generals cannot be solved



- Faulty processes are shown coloured
- Notation: “:” reads “says”.
  E.g.: “3:1:u” is the message “3 says 1 says u”
- From the point of view of p_2, the two situations are identical
- Any algorithm would take the wrong decision in one situation!

Satoshi’s solution: Proof of Work for verifying transactions

- **In blockchains, the data to agree on is the history of transactions**
  - Dishonest nodes can claim different transactions to different nodes
- Lamport’s result means that, with no further assumptions, 33% dishonest nodes can destroy the system (instead of the majority).
- **The problem is that dishonest nodes can too easily generate incoherent confirmations!**
- Satoshi’s idea: **make verification and announcements HARD, so that being dishonest becomes more expensive than being honest!**

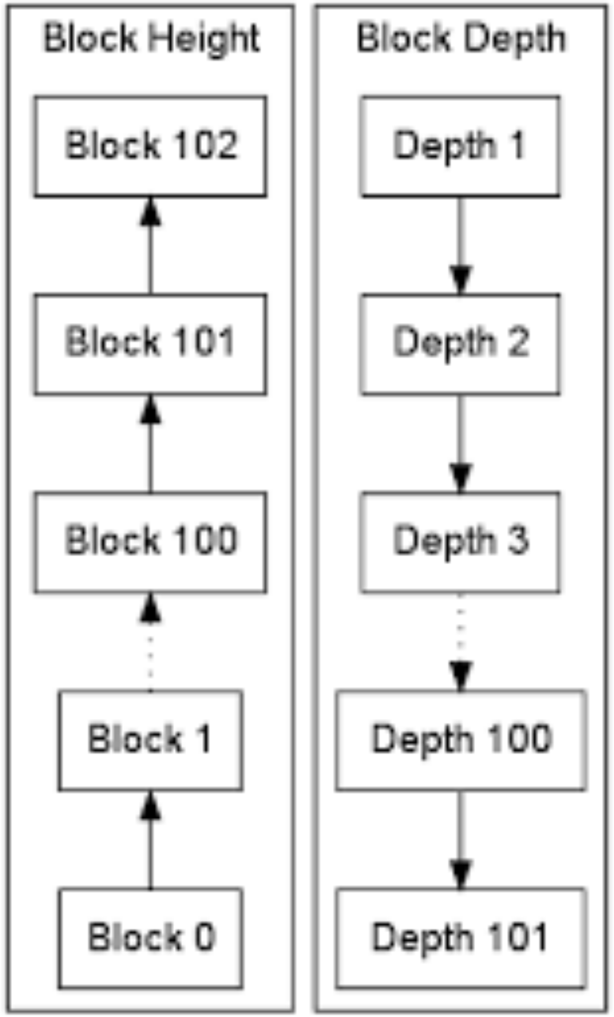
Verifying transactions in a Proof-of-Work system

If Dave wants to validate a bunch of transactions (a *block*), he has to:

1. Check transactions against his copy of the blockchain to make sure are legitimate (like before)
2. Spend (lots of) resources to solve a **hard mathematical puzzle (proof-of-work)** (in Bitcoin, this is called **mining**)
3. **Only after he has found the solution**, he can announce it to the network, together with the block of verified transactions.
4. Every nodes receives the block, checks that the transactions are legitimate and Dave has actually found a solution for the puzzle. If everything is fine, it add the block to the blockchain
5. Dave gains a reward (thus Proof-of-work is a **competition)**

Validated blocks = the Blockchain

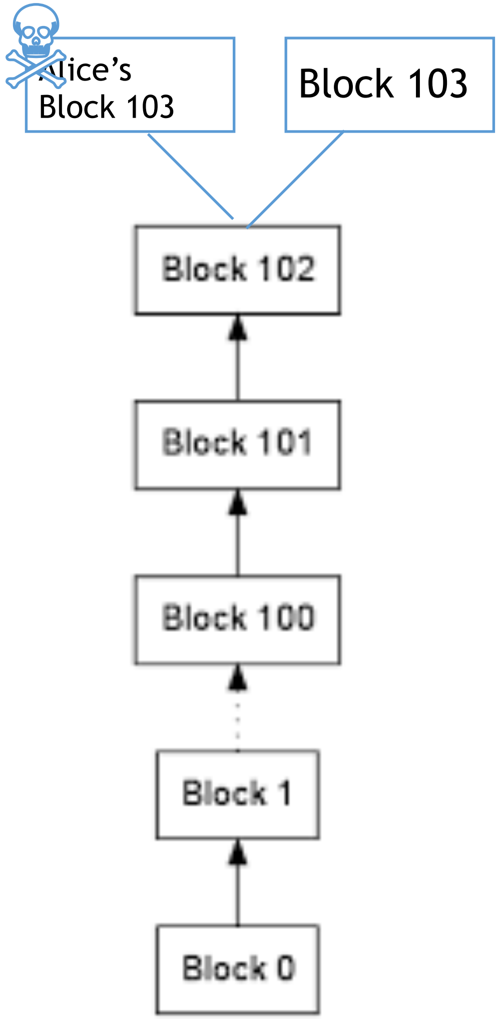
- Each time a valid block is found by some miner, it is broadcast to the whole network, and each node add it to its copy of the ledger
- Each block references a previous block, hence the whole set of blocks is called **blockchain**
- Not all nodes are involved in mining: most nodes just wait for the others to solve the puzzle and announce the block
    - These nodes just keep a local copy of the blockchain.



Block Height Compared
To Block Depth

Why Proof of work prevents bad behaviours??

- Suppose a dishonest group of miners (Alice's gang) tries to announce a new block, possibly containing some false data, to the current blockchain
  - To achieve this, they have to "win" the puzzle race
  - But **the likelihood of being first to solve the puzzle is proportional to the collective computational power put into the search!**
  - If 51% of the overall computational power on the network is controlled by honest miners, it is more likely that some honest node will win the race before Alice's gang
  - In this case, a correct block will be added to the blockchain instead of Alice's bad one, and Alice's gang has to start over from the new block!



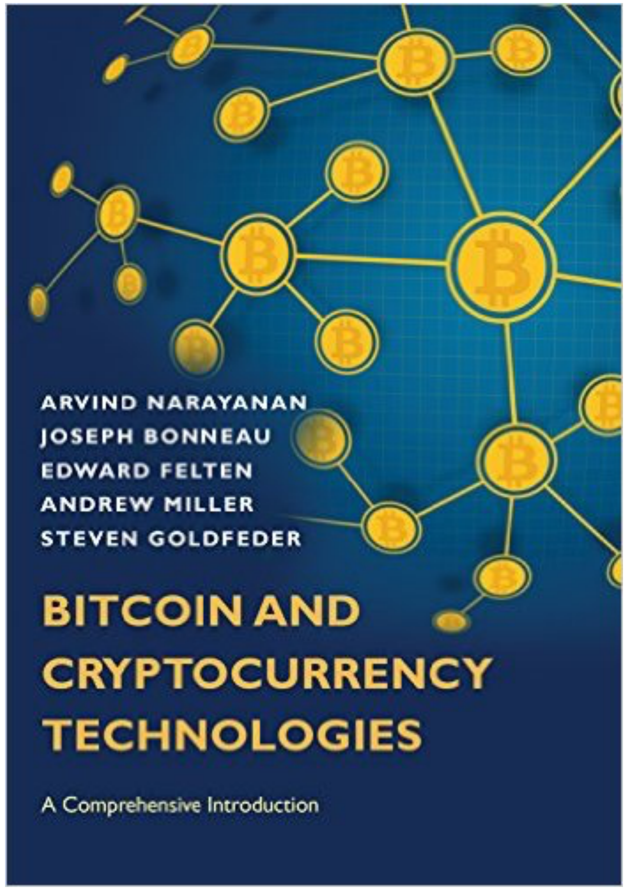
Satoshi’s agreement: the *computational* majority is always right

- Although forks may happen, eventually the blockchain will contain only blocks mined by the computational majority of the network
- Put as a “democratic” principle:

- This form of eventual agreement is called **Satoshi’s agreement**
- A block (and the transactions therein) can be retracted after it has been announced because a longer chain has been discovered, but the older is the block, the more unlikely this happens

For a deeper understanding

- **"Bitcoin and Cryptocurrency Technologies"** excellent book (from where several pictures of this presentation have been taken)
- Available at http://bitcoinbook.cs.princeton.edu



To conclude

- Blockchains are a good example of how Computer Science can affect people, society, economics, politics…
- But also: AI, Cryptography, Fintech, Elections… the impact of CS cannot be overestimated
- This is why it is important to know Computer Science

> *A handful of people, having no relation to the will of society, having no communication with the rest of society, will be taking decisions in secret which are going to affect our lives in the deepest sense* (C. P. Snow, 1961)

Are you going to be among those who decide, or among those who are affected?

Thank you for your attention

Any Question?

marino.miculan@uniud.it