Confiabilidade de Sistemas Distribuídos
Dependable Distributed Systems

DI-FCT-UNL, Nuno Preguiça

Lect. 4
BlockChain and Bitcoin

2018/2019, 2nd SEM

MIEI
Mestrado Integrado em Engenharia Informática

Last lecture: Byzantine fault model

- Processes that fail can exhibit arbitrary behavior
  - Return wrong replies
  - Take too long to execute a computation step
  - Do not follow the communication protocol
  - Collude with other processes

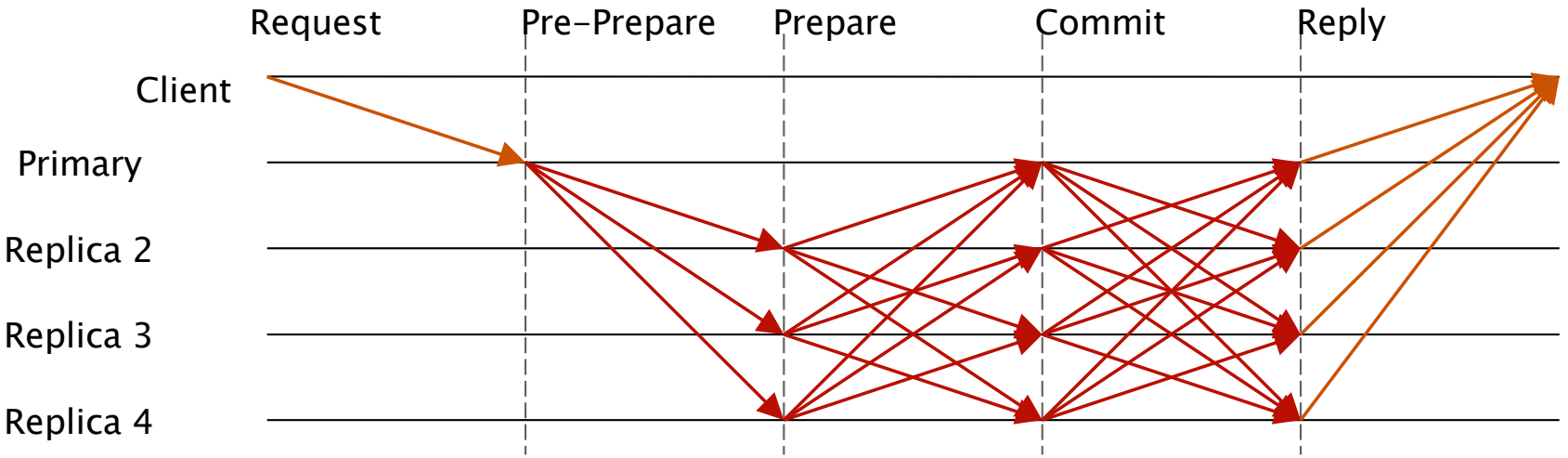
Byzantine fault tolerance

- Correctness property for replicated system
- The behavior of a system must be equivalent to that of a non-replicated system where operations execute instantaneously in a moment between the moment the operation is invoked and the result is returned

Byzantine read/write register replication

- Byzantine quorums
  - With signatures: N = 3f+1, Q=2f+1
  - Without signatures: N = 4f+1, Q=3f+1
- ABD algorithm for BFT

State-machine BFT replication



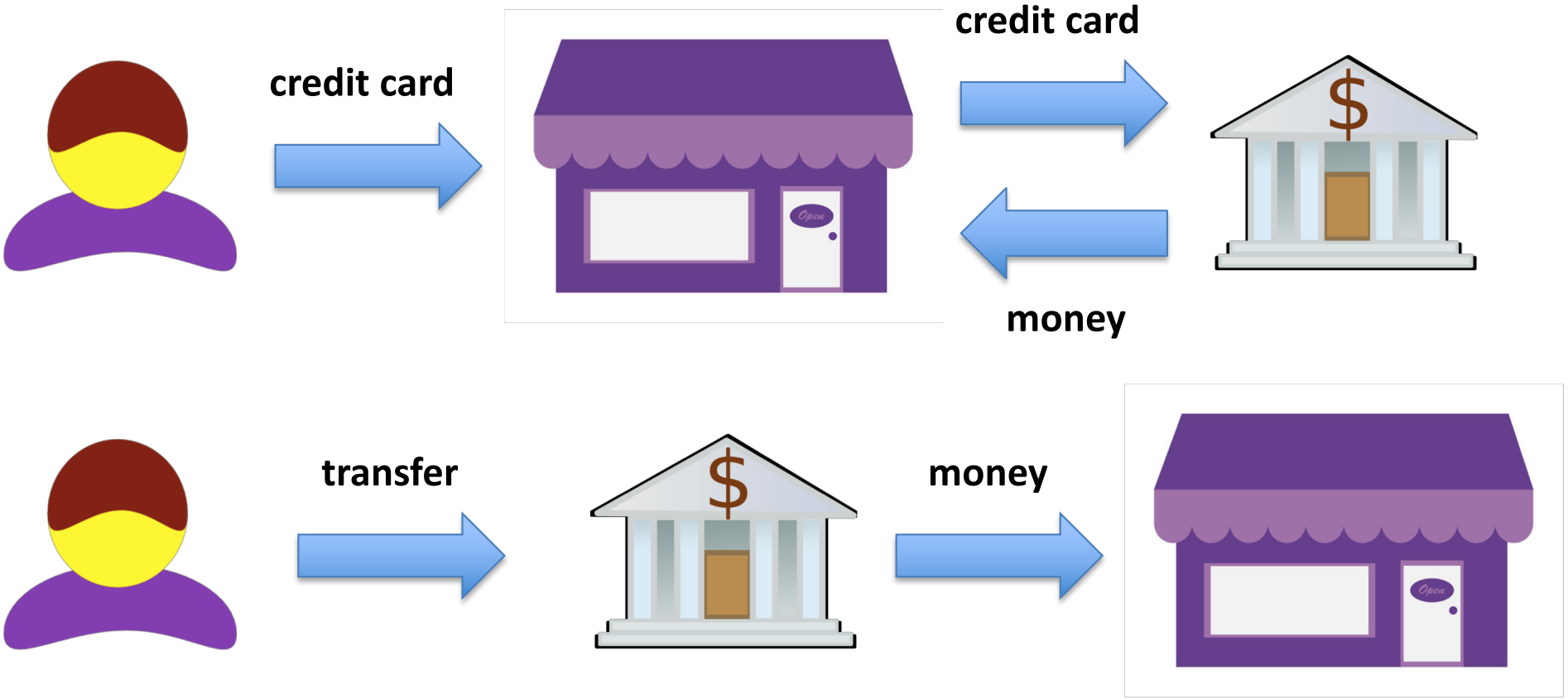
- Replication algorithm that tolerates Byzantine faults
  - State-machine replication
    - The same sequence of operations is executed in all replicas
    - Guarantees that all coorrect replicas will converge to the same state

Today's lecture

- Decentralized BFT state machine database
  - Bitcoin and blockchain

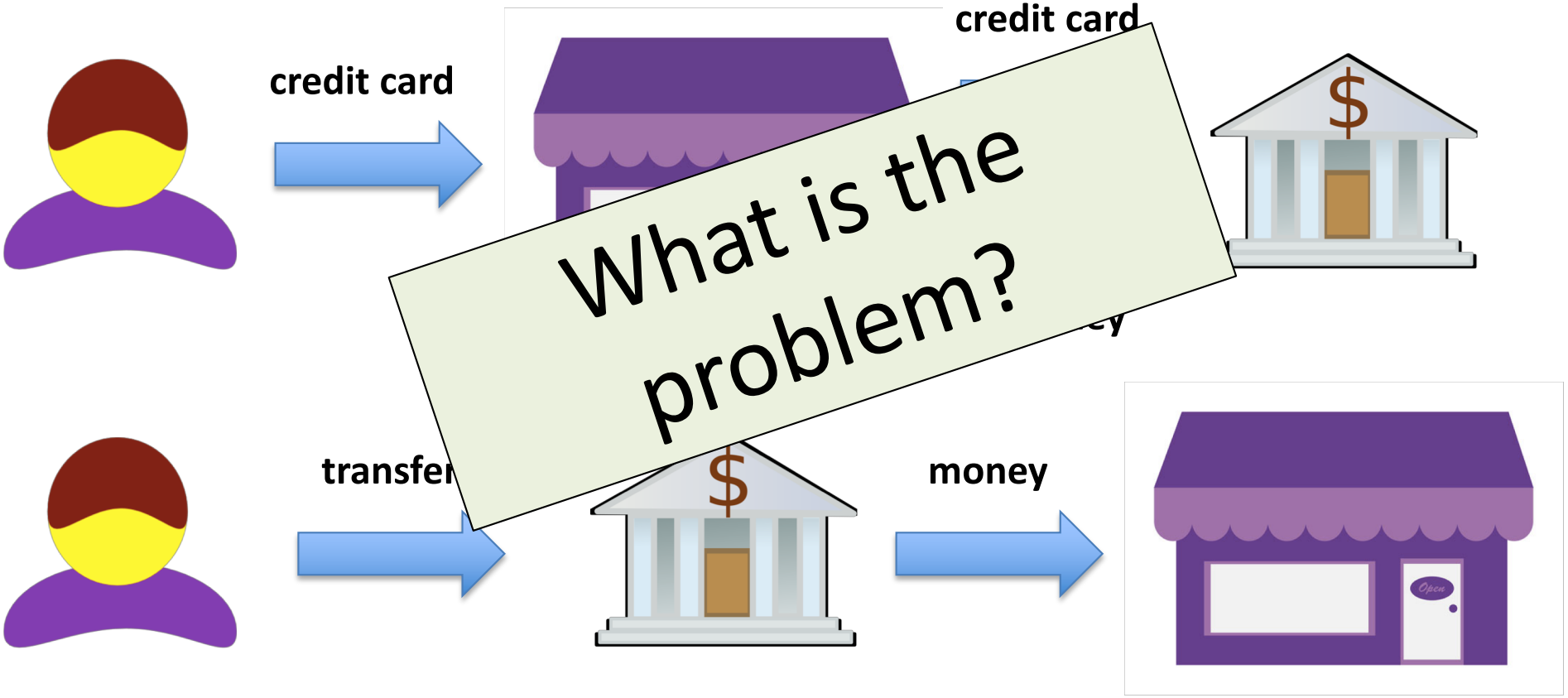
Context

- Traditional systems for payment



Context

- Traditional systems for payment



Problems of traditional payment systems

- Intermediaries get a fee
  - Necessary for supporting the payment system, inherent frauds
  - Might be unreasonable for small payments
- Process is slow
  - Particularly when involving multiple entities – e.g. bank transfers
- Payments cannot be anonymous
  - Regulations

Goal

- Payment systems = mechanism for supporting money transfers
  - Quick
  - Anonymous
  - Low transfer fee
  - No double spending
  - No central authority

Bitcoin

- Cryptocurrency introduced by “Satoshi Nakamoto” in 2008
- currency unit: Bitcoin (BTC) 1 BTC = 10^8 Satoshi

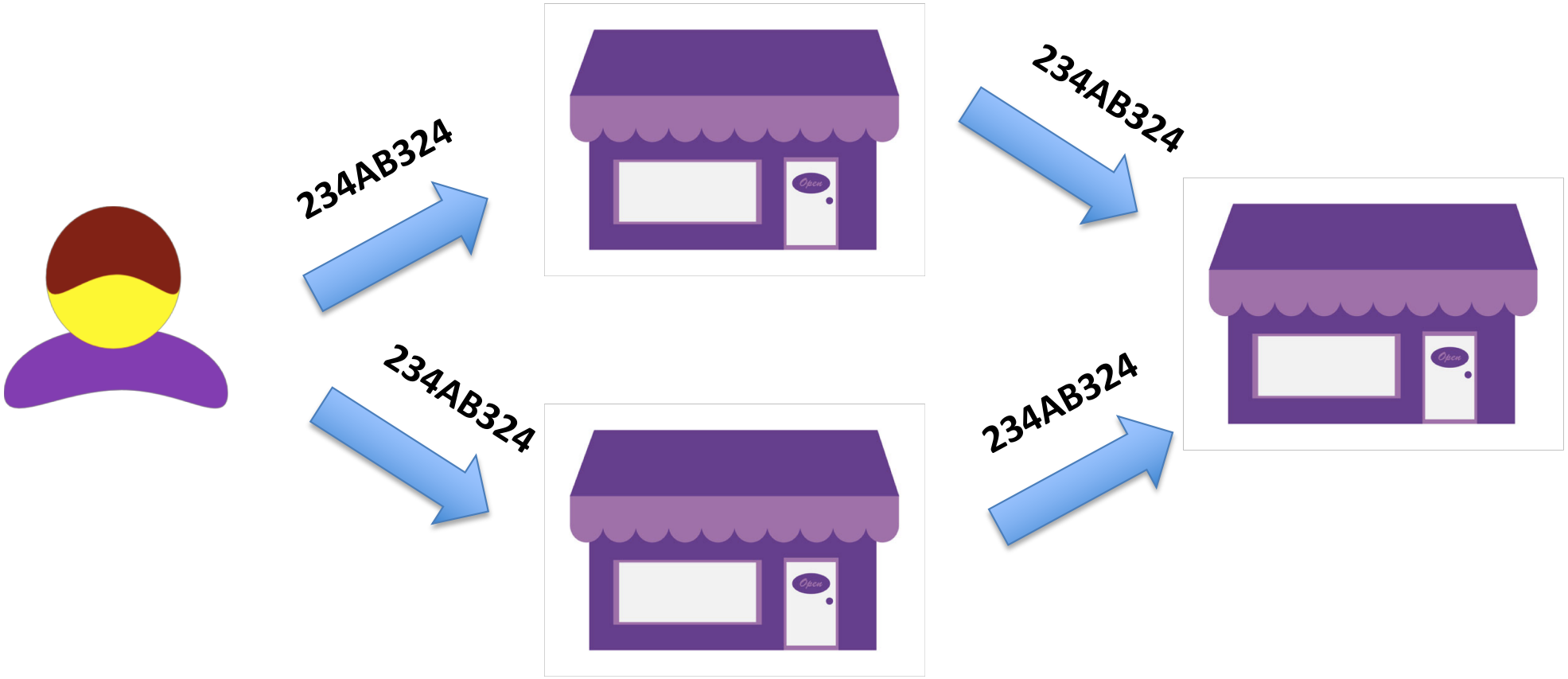
Bitcoin = other currency?

- **Bitcoin** value comes from the fact that people expect other people will accept it in the future

- **Plus**
  - The maximum amount of money is fixed (around 21 M BTC)
    - No central bank can emit money => no inflation
- **Minus**
  - Assumption for value similar to other currencies, but not guaranteed by any country

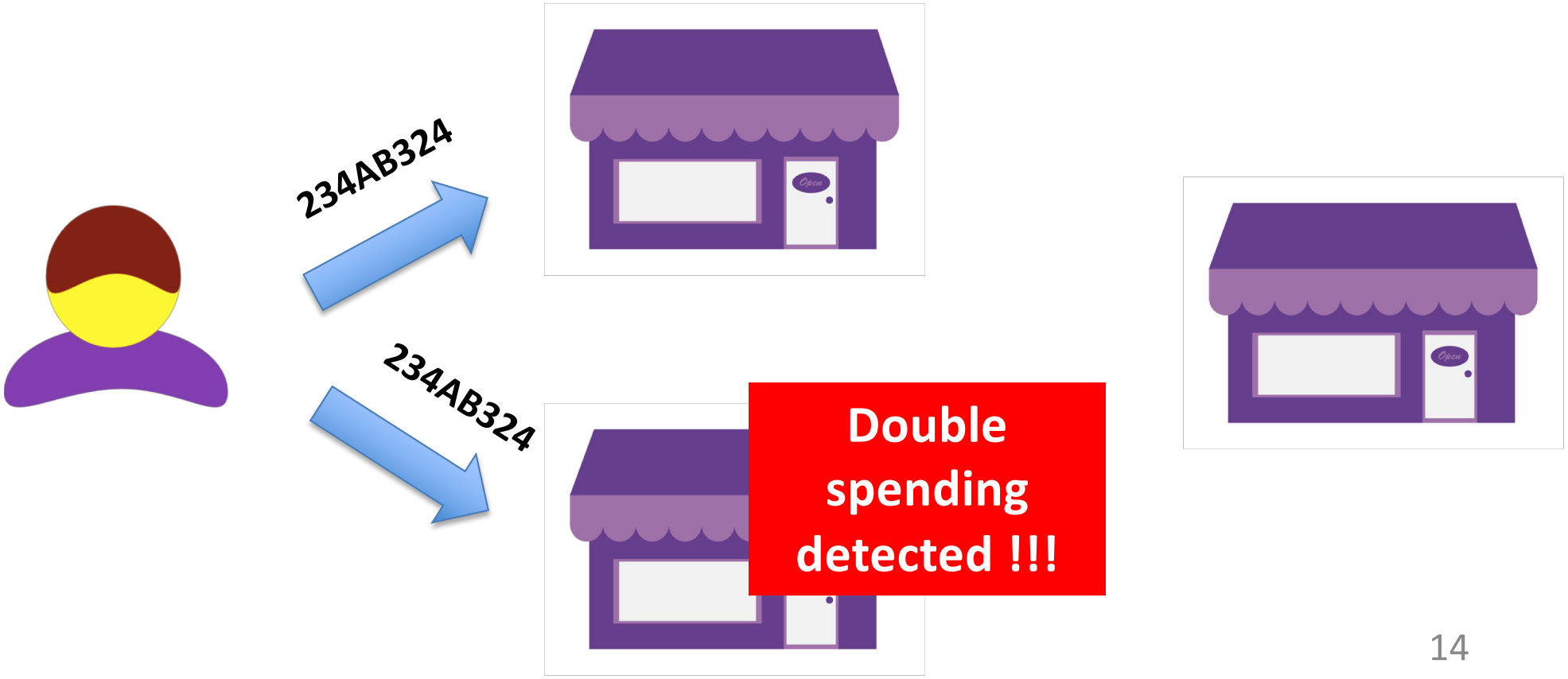
Digital currencies: main problem

- Avoid double spending



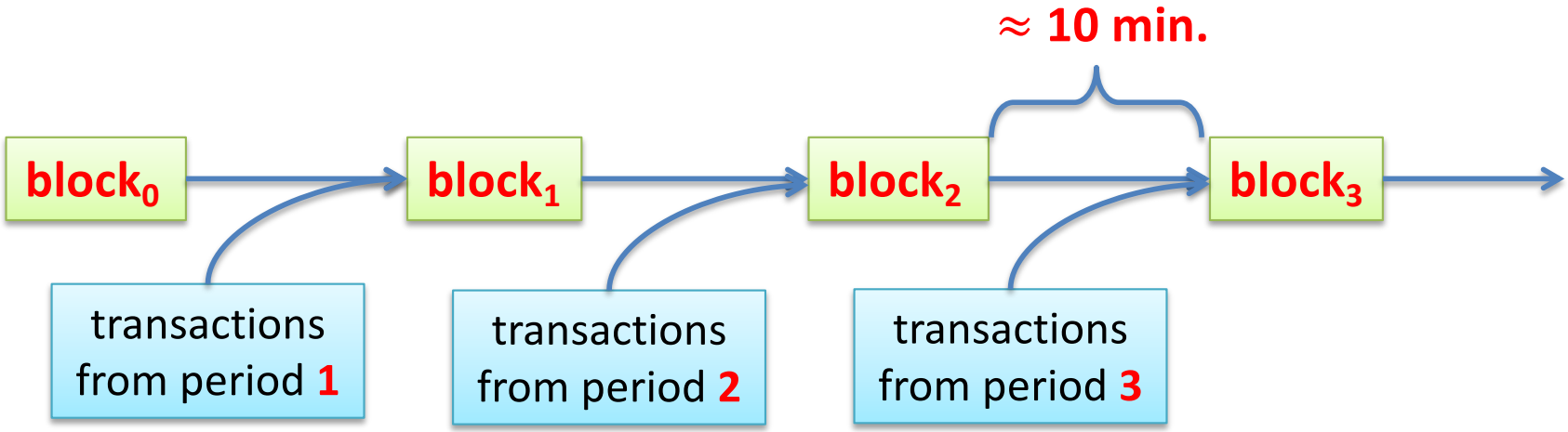
Avoiding double spending

- Bitcoin: keep a public registry of transfers to detect double spending before it is too late



Public registry: block chain

- Database maintained by the participants in the system
  - Sequence of blocks, each one maintaining a list of transactions



Note: the ideas presented are those of Bitcoin, but some details are not exactly the same

Proving ownership... and anonymously

- Rely on public key cryptography
- Transfer record for: **A** wants to transfer **m** to **B**
  - Entry in the log: $[KpubA,KpubB,m]_{KprivA}$
- Anonymous because new keys can be generated for every transaction
- Someone with the full list of transfers can verify that A owns that money
- $[KpubA,KpubB,m]_{KprivA}$ = (A,KpubB,m) signed with the private key of A

Proving ownership... and anonymously

- Rely on public key cryptography
- Transfer re
  - Entry in
- Anonymo
  every transaction
- Someone with the full list of transfers can verify that A owns that money

Why would someone keep the full list of transfers for verifying the validity of transfers?

How to guarantee that participants agree in the order of the list?

- $[KpubA,KpubB,m]_{KprivA}$ = (A,KpubB,m) signed with the private key of A

Incentives for verifiers (miners)

- Whenever a transaction is verified and added to the log, miners are paid
  - More precisely:
    - a miner verifies a batch of transactions stored in a block
    - whenever a new block is generated, the miner gets new bitcoins (currently: 12.5 BTC)
    - the system is designed and continuously adjusted to guarantee that a new block is generated around once every 10 minutes

What is involved in creating a new block?

1. Get a set of pending transactions (typically prioritizing those with fees)

2. Verify the transactions
   - To guarantee the soundness of the system

3. Solve a hashing problem
   - To guarantee that a new block is created only every 10 minutes

How to verify a transaction?

- **A** wants to transfer **m** to **B**
  - Entry in the log: $[KpubA,KpubB,m]_{KprivA}$

- Verify the signature of the transfer
- Check that A has not transferred the money before
  - Blocks need to form a totally ordered list
    - Each block has an hash of the previous record, making it impossible for an attacker to replace an old block

The Hashing Problem

- A new block contains
  - hash of previous block
  - new transactions
  - creation of reward bitcoins
  - nonce

- A block is valid if the hash of the block ends with enough zeros, as determined by the current difficulty. Complexity for 1, 2, … zeros?
- Miner must try different nonce's until it find one with the required properties

Proving ownership... and anonymously

- Rely on public key cryptography
- Transfer re
  - Entry in
- Anonymo
  every transaction
- Someone with the full list of transfers can verify that A owns that money

Why would someone keep the full list of transfers for verifying the validity of transfers?

How to guarantee that participants agree in the order of the list?

- $[KpubA,KpubB,m]_{KprivA}$ = (A,KpubB,m) signed with the private key of A

Consensus in Bitcoin

- Multiple miners are trying to generate the next block, N
- Whenever a miner solves the hashing problem for block N, it makes the new block public
  - Other miners stop trying to generate block N and start trying to find block N+1

What if the network gets partitioned or multiple miners find a block concurrently?

Consensus in Bitcoin (2)

- The system is designed to keep always the longer blockchain
- If different blocks N are announced, miners will start generating a new block N+1 based on each of the announced blocks
- The first block N+1 announced will decide which block N survives

- When can a transfer be considered stable?

Consensus in Bitcoin (3)

- A transaction is said to have received k confirmations if it has been published in a block that has been added to the blockchain, and k-1 more blocks have also been added
- A transactions is typically considered “confirmed” once it has 6 confirmations.
- Newly minted Bitcoins are typically considered confirmed once they have received 100 confirmations.

Security

- An attacker cannot spend money from other users. Why?
- An attacker can try to:
  - Generate new blocks quicker than other nodes
    - This would require the attacker to have more computing power than the rest of the network
  - Double-spend
    - This would require the attacker to generate blocks with invalid transfers.
    - As a transfer is only considered stable after K confirmations, this would require the attacker to be able to generate new blocks quicker than other nodes

Bitcoin scalability challenges

[from On Scaling Decentralized Blockchains]

- Maximum throughput: 3.3-7 tx/s
- Cost per Confirmed Transaction (CPCT): USD$6

- How to scale Bitcoin?

To know more

- Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. 2008; https://bitcoin.org/bitcoin.pdf

- Croman, K., et. Al. On Scaling Decentralized Blockchains (A Position Paper). Proc International Conference on Financial Cryptography and Data Security 2016. http://fc16.ifca.ai/bitcoin/papers/CDE+16.pdf

- Sarah Underwood. 2016. Blockchain beyond bitcoin. *Commun. ACM* 59, 11 (October 2016), 15-17.
- http://cacm.acm.org/magazines/2016/11/209132-blockchain-beyond-bitcoin/fulltext